EMBEDDIA

Cross-Lingual Embeddings for Less-Represented Languages in European News Media

Research and Innovation Action
Call: H2020-ICT-2018-1
Call topic: ICT-29-2018 A multilingual Next generation Internet
Project start: 1 January 2019 Project duration: 36 months

D2.3: Initial keyword extraction techniques (T2.2)

Executive summary

The task of multilingual keyword extraction and matching (Task T2.2) addresses the problem of extracting topical terms and keywords from the input text in a monolingual and multilingual problem setting, aiming to detect keywords and terms applicable in news linking, cross-lingual topic modelling and news analysis in later stages of the EMBEDDIA project. This deliverable addresses keyword extraction state-of-the art, evaluation methodology, as well as novel keyword extraction methods, including an unsupervised graph-based technique, and a supervised method based on transformer neural networks, and an application of selected methods to the Croatian datasets of news in EMBEDDIA. In addition, we present our work on term extraction and alignment, where we conducted a reproducibility study of a bilingual terminology alignment approach, developed new bilingual terminology extraction approaches, and tested how embeddings can be used as a terminology expansion technique.

Partner in charge: JSI

Project co-funded by the European Commission within Horizon 2020 Dissemination Level		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	–
RE	Restricted to a group specified by the Consortium (including the Commission Services)	–
CO	Confidential, only for members of the Consortium (including the Commission Services)	–

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825153

Deliverable Information

Document administrative information	
Project acronym:	**EMBEDDIA**
Project number:	**825153**
Deliverable number:	**D2.3**
Deliverable full title:	**Initial keyword extraction techniques**
Deliverable short title:	**Initial keyword extraction**
Document identifier:	**EMBEDDIA-D23-InitialKeywordExtraction-T22-submitted**
Lead partner short name:	**JSI**
Report version:	**submitted**
Report submission date:	**31/12/2019**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Senja Pollak (JSI), Matej Martinc (JSI), Andraž Repar (JSI), Blaž Škrlj (JSI)**
Co-author(s):	**Nada Lavrač (JSI), Nika Eržen (JSI)**
Status:	**__ draft, __ final, _x_ submitted**

The EMBEDDIA Consortium partner responsible for this deliverable has addressed all comments received. Changes to this document are detailed in the change log table below.

Change log

Date	Version number	Author/Editor	Summary of changes made
01/11/2019	v0.1	Andraž Repar (JSI)	Report structure.
15/11/2019	v0.2	Senja Pollak, Nada Lavrač (JSI)	First draft.
18/11/2019	v0.3	Matej Martinc, Blaž Škrlj, Nika Eržen (JSI)	Supervised and unsupervised keyword extraction approaches.
21/11/2019	v0.4	Andraž Repar (JSI) (JSI)	Introduction and terminology extraction and alignment parts.
22/11/2019	v0.5	Senja Pollak, Matej Martinc, Nada Lavrač (JSI)	Final version for internal review.
28/11/2019	v0.6	Matthew Purver (QMUL)	Internal review; comments added.
28/11/2019	v0.7	Nada Lavrač (JSI)	Minor revision of appendices.
01/12/2019	v0.8	Shane Sheehan (UEDIN)	Internal review; comments added.
08/12/2019	v0.9	Senja Pollak, Andraz Repar (JSI)	First part of revisions implemented.
12/12/2019	v0.10	All JSI members	Second part of revisions implemented.
18/12/2019	prefinal	Nada Lavrač (JSI)	Report quality checked and finalised.
23/12/2019	final	Senja Pollak (JSI)	Minor corrections.
23/12/2019	submitted	Tina Anžič (JSI)	Report submitted.

Table of Contents

List of abbreviations

BTA	Bilingual Terminology Alignment
CRF	Conditional Random Fields
DoA	Description of Action
EC	European Commission
LSTM	Long Short Term Memory
ATE	Automatic Term Extraction
ATR	Automatic Term Recognition

1 Introduction

This deliverable reports on the initial results achieved in multilingual keyword extraction and matching within Task T2.2 of the EMBEDDIA project, which lasts for 2 years, described in the EMBEDDIA Description of Action (DoA) as follows:

> We will use monolingual and multilingual methods to extract topical terms and keywords from the text. We will apply and further develop our statistical approaches (based on heuristics), machine learning approaches, as well as graph-based approaches. We will test how keyword extraction helps in anchoring of embeddings and improving cross-lingual word embeddings (see WP1). We will also use cross-lingual word embeddings to detect keywords and terms to be used in cross-lingual topic modelling. Structured knowledge resources, including ontologies will be embedded into vector space and used as a background knowledge to improve the extraction.

Specifically, this deliverable at M12 describes our research on keyword extraction and the closely related topic of terminology extraction, as well as cross-lingual term alignment, which we plan to adapt for multilingual keyword matching.

Let us start by defining the terms used in this report.

Keywords are terms (i.e. expressions) that best describe the subject of a document (Beliga et al., 2015) and a good keyword effectively summarises the content of a document allowing it to be efficiently retrieved when needed.

Terms are verbal designations of general concepts in a specific subject field (as defined in ISO 1087 standard). In contrast to keywords, which are usually assigned on a single document level, terms are more frequently used on a document collection level, i.e. domain level. While not all terms are keywords, there is a strong overlap between the most frequent terms and keywords, therefore term extraction techniques can be applied successfully to keyword extraction.

Named Entities include person names, locations, organisations etc. (Hoffart et al., 2011). While they may assume the role of keywords, named entity recognition is a distinct research area and is not part of this deliverable, but is reported in deliverable D2.2.

Keyword extraction refers to the process of extracting keywords from documents.

Monolingual terminology extraction refers to the process of finding terms within a collection of documents from a specific subject field.

Bilingual terminology extraction is the process which, given the input of related specialized monolingual corpora, results in the output of terms aligned between two languages.

Bilingual terminology alignment is the process of aligning terms between two candidate term lists in two languages.

Term expansion is the process of extending a list of existing terms by novel term candidates.

In a media analysis setting, keywords correspond to tags that are added to articles by news providers (e.g., EMBEDDIA news media partners). On the other hand, terms can be understood as expressions characteristic of different news categories (e.g., sports vs. foreign policy). As similar techniques can be used for both, the field has a strong exploitation potential for applying methods from the media setting (keywords) to the translation industry and terminography (terms), and vice versa.

In this work, we present the state-of-the art, as well as novel methods for keyword extraction, including a novel unsupervised graph-based technique, and a novel supervised method based on transformer neural networks. Selected methods are also applied to the Croatian datasets of news gathered within EMBEDDIA. In addition, we report our work on term extraction and alignment, where we conducted a reproducibility study of a bilingual terminology alignment approach (that can be adapted in future to cross-lingual keyword matching), created bilingual terminology extraction approaches, and tested how embeddings can be used as a terminology expansion technique.

The work performed in the scope of task T2.2 resulted in several papers, which are included in the Appendices of this deliverable:

- **Škrlj et al. (2019)**: "Rank-based Keyword extraction via Unsupervised learning and Meta vertex aggregation", presented at Statistical Language and Speech Conference (2019), which introduces a novel unsupervised keyword extraction technique. (Appendix A)
- **Repar, Martinc, & Pollak (2019)**: "Reproduction, replication, analysis and adaptation of a term alignment approach", published in Language Resources and Evaluation journal, which reimplements and adapts an approach to link two lists of terms (or keywords) (Appendix B).
- **Repar, Podpečan, et al. (2019)**: "TermEnsembler: An ensemble learning approach to bilingual term extraction and alignment", published in the journal Terminology, which proposes a workflow for bilingual terminology extraction, consisting of a statistical method for extracting terms (or keywords) and a combination of term-alignment techniques (Appendix C).
- **Pollak et al. (2019)**: "Karst Exploration: Extracting Terms and Definitions from Karst Domain Corpus," presented at eLex 2019 conference, and introducing a domain modelling pipeline, where term expansion by embedding is the most relevant to EMBEDDIA (Appendix D).

Note that we plan to submit another paper on novel supervised method for keyword extraction TNT-KID, described in this deliverable, after finalising the experiments.

This deliverable is structured as follows. Section 2 presents the state-of-the-art. It starts by outlining the related keyword extraction research, evaluation methodology and the datasets used in this work in Section 2.1, followed by the related work in terminology extraction and alignment in Section 2.2. Section 3 presents our work in the field of keyword extraction. Section 3.1 presents our attempts to reproduce the results of the unsupervised state-of-the-art keyword extraction method. Next, we introduce novel methods for keyword extraction: a graph-based unsupervised method is presented in Section 3.2, while Section 3.3 reports on our promising supervised approach called TNT-KID; a transformer-based neural tagger for keyword identification. We compare the results of our and state-of-the-art supervised and unsupervised approaches in Section 3.4. The final experiments related to keywords consist of testing selected approaches on Croatian EMBEDDIA datasets in Section 3.5. Section 4 presents our work in bilingual terminology extraction and alignment. In Section 4.1 we present a study in which we reimplemented and adapted an approach to term alignment. Next, Section 4.2 reports on TermEnsembler, an ensemble learning approach to bilingual term extraction and alignment. Section 4.3 reports on our experiments with graph-based term alignment using co-frequency information from a bilingual parallel corpus. Section 4.4 contains a description of a domain modelling experiment, including term extraction and term expansion. Section 5 contains the links to associated outputs (source code), followed by the conclusions and presentation of plans for future work. The related published papers are added in Appendices A–D.

2 Background and related work

This section presents related work in keyword extraction in Section 2.1, followed by the related work in bilingual terminology extraction and alignment in Section 2.2.

2.1 Related work on keyword extraction

This section first overviews selected methods in keyword extraction (supervised in Section 2.1.1 and unsupervised in Section 2.1.2). Selected datasets used for state-of-the-art comparisons are reported in Section 2.1.3. The final section presents the evaluation measures used in keyword extraction tasks in Section 2.1.4.

2.1.1 Supervised keyword extraction methods

Traditional supervised approaches to keyword extraction considered the task as a two step process (the same is true for unsupervised approaches). First, a number of syntactic and lexical features are used to extract keyword candidates from the text. Secondly, the extracted candidates are ranked according to different heuristics and the top n candidates are selected as keywords (Yuan et al., 2019). One of the first supervised approaches to keyword extraction was proposed by Witten et al. (2005), whose algorithm named KEA uses only term frequency - inverse document frequency (TF-IDF) and the terms position in the text as features for term identification. These features are fed to the Naive Bayes classifier, which is used to determine for each phrase in the text if it is a keyword or not. Medelyan et al. (2009) managed to build on the KEA approach and proposed the *Maui* algorithm, which also relies on the Naive Bayes classifier for candidate selection but employs additional semantic features, such as e.g., *node degree*, which quantifies the semantic relatedness of a candidate to other candidates, and *Wikipedia-based keyphraseness*, which is the likelihood of a phrase being a link in the Wikipedia corpus.

A more recent supervised approach is a so-called sequence labelling approach to keyword extraction by Gollapalli et al. (2017), where the idea is to train a keyword tagger using token-based linguistic, syntactic and structural features. The approach relies on a trained Conditional Random Field (CRF) tagger and the authors demonstrated that this approach is capable of working on-par with slightly older state-of-the-art systems that rely on information from Wikipedia and citation networks, even if only within-document features are used. Another sequence labeling approach proposed by Luan et al. (2017) builds a sophisticated neural network by combing an input layer comprising a concatenation of a word, character and part-of-speech embedding, a bidirectional Long Short-Term Memory (LSTM) layer and a CRF tagging layer. They also propose a new semi-supervised graph based training regime for training the network.

The newest state-of-the-art approaches to keyword detection consider the problem as a sequence-to-sequence generation task. The first research leveraging this tactic was proposed by Meng et al. (2017), employing a generative model for keyword prediction with a recurrent encoder-decoder framework with an attention mechanism capable of detecting keywords in the input text sequence and also potentially finding keywords that do not appear in the text. Since finding absent keywords involves a very hard problem of finding a correct class in a set of usually thousands of unbalanced classes, their model also employs a copying mechanism (Gu et al., 2016) based on positional information, in order to allow the model to find important keywords present in the text, which is a much easier problem.

Very recently, the model proposed by Meng et al. (2019) has been somewhat improved by investigating different ways in which the target keywords can be fed to a classifier during the training phase. While the original system used a so-called *one-to-one* approach, where a training example consists of an input text and a single keyword, the improved model now employs a *one-to-seq* approach, where an input text is matched with a concatenated sequence made of all the keywords for a specific text. The study also shows that the order of the keywords in the text matters. A *one-to-seq* approach has been even further improved by Yuan et al. (2019), who incorporated two diversity mechanisms into the model. The mechanisms (called *semantic coverage* and *orthogonal regularization*) constrain the over-all inner representation of a generated keyword sequence to be semantically similar to the overall meaning of the source text and therefore force the model to produce diverse keywords.

The neural sequence-to-sequence models are capable of outperforming all older supervised and unsupervised models by a large margin but do require a very large training corpora with tens of thousands of documents for successful training. This means that their use is limited only to languages (and genres) in which large corpora with manually labeled keywords exist. In this deliverable we present our work on developing a novel neural approach TNT-KID in Section 3.3.

2.1.2 Unsupervised keyword extraction methods

While the previous section discussed recently emerged methods for keyword extraction that operate in a supervised learning setting, supervised learning can be data-intensive and time consuming. *Unsupervised* keyword detectors can tackle these two problems, yet at the cost of reduced overall performance.

Unsupervised approaches need no training and can be applied directly without relying on a gold standard document collection. They can be divided into statistical and graph-based methods.

- Statistical methods, such as KP-MINER (El-Beltagy & Rafea, 2009), RAKE (Rose et al., 2010) and YAKE (Campos et al., 2018a,b), use statistical characteristics of the texts to capture keywords.
- Graph-based methods, such as TextRank (Mihalcea & Tarau, 2004), Single Rank (Wan & Xiao, 2008), TopicRank (Bougouin et al., 2013) and Topical PageRank (Sterckx et al., 2015), build graphs to rank words based on their position in the graph.

Among the statistical approaches, the state-of-the-art keyword extraction algorithm is YAKE (Campos et al., 2018a,b), which is also one of the best performing keyword extraction algorithms overall; it defines a set of five features capturing keyword characteristics which are heuristically combined to assign a single score to every keyword.

On the other hand, among the graph-based approaches, Topic Rank by Bougouin et al. (2013) can be considered state-of-the-art; candidate keywords are clustered into topics and used as vertices in the final graph, used for keyword extraction. Next, a graph-based ranking model is applied to assign a significance score to each topic and keywords are generated by selecting a candidate from each of the top-ranked topics. Network-based methodology has also been successfully applied to the task of topic extraction (Spitz & Gertz, 2018). In this deliverable (see Section 3.2), we describe RaKUn (Škrlj et al., 2019), a novel unsupervised, language-agnostic graph-based method that explores how vertices can be *aggregated* prior to keyword detection.

2.1.3 Selected keyword extraction datasets

In deliverable D2.1, the list of selected datasets was longer, but the subselection below, used in experiments in this deliverable, is chosen in order to allow for comparison with best supervised and unsupervised approaches from related work. The statistics about the datasets used are presented in Table 1. Evaluation measures discussed in Section 2.1.4 are used for comparing the performance of different algorithms, where the performance is measured on six distinct datasets:

- **KP20k (Meng et al., 2017)**: This dataset contains titles, abstracts, and keyphrases of 40,000 scientific articles from the field of computer science. Half of these articles (20,000) are used as a test set and 20,000 are used as train set.
- **Inspec (Hulth, 2003)**: The dataset contains 2,000 abstracts of scientific journal papers in computer science. Two sets of keywords are assigned to each document, the controlled keywords that appear in the Inspec thesaurus, and the uncontrolled keywords, which are assigned by the editors. Only uncontrolled keywords are used in the evaluation, same as by Meng et al. (2017), and the dataset is split into a 500 test papers and 1500 train papers.
- **Krapivin (Krapivin et al., 2009)**: This dataset contains 2,304 full scientific papers published by ACM with author-assigned keyphrases. 400 papers from the dataset were used as a test set and the others are used for training.
- **NUS (Nguyen & Kan, 2007)**: The dataset contains 211 scientific conference papers and contains a set of keywords assigned by student volunters and a set of author assigned keywords, which are

both used in evaluation, where 20% of papers were randomly selected for a test set, others were used for training.

- **SemEval (Kim et al., 2010)**: The dataset used in the SemEval-2010 Task 5, Automatic Keyphrase Extraction from Scientific Articles, contains 288 articles collected from the ACM Digital Library. 100 articles were used for testing and the rest were used for training.
- **DUC (Wan & Xiao, 2008)**: The dataset consists of 308 English news articles and contains 2,488 hand labeled keyphrases, where 20% of articles were randomly selected for a test set, others were used for training.

2.1.4 Evaluation measures

In information extraction, there are several evaluation metrics that can be used to measure the performance of models and compare them with the state-of-the-art. The measures include precision@k, recall@k, F1@k, precision@$\mathcal{O}$, recall@$\mathcal{O}$ and F1@$\mathcal{O}$:

Precision@k. In a ranking task, we are interested in precision at rank k. This means that only the keywords ranked equal to or higher than k are considered and the rest are disregarded. Precision is the ratio of the number of relevant keywords divided by the number of keywords returned by the system.

$$precision = \frac{|relevant \quad keywords@k|}{|returned \quad keywords|} \tag{1}$$

Recall@k. Recall@k is the ratio of the number of relevant keywords ranked equal to or higher than k by the system divided by the number of correct ground truth keywords.

$$recall = \frac{|relevant \quad keywords@k|}{|correct \quad keywords|} \tag{2}$$

Due to the high variance of a number of ground truth keywords, this type of recall becomes problematic if k is smaller than the number of ground truth keywords, since it becomes impossible for the system to achieve a perfect recall. (Similar can happen to precision@k, if number of keywords in a gold standard is lower than k, and returned number of keywords is fixed at k.)

F1@k is a harmonic mean between Precision@k and Recall@k.

$$F1@k = 2 * \frac{P@k * R@k}{P@k + R@k}$$

Precision@$\mathcal{O}$. Here, $\mathcal{O}$ denotes the number of ground truth keyphrases. This means that only the keywords ranked higher or equal than $\mathcal{O}$ are considered and the rest are disregarded.

$$precision = \frac{|relevant \quad keywords@\mathcal{O}|}{|returned \quad keywords|} \tag{3}$$

Table 1: Datasets used for empirical evaluation of keyword extraction algorithms.

Dataset	lang	No. docs	Avg. keywords	Avg. doc length	% Present keywords
Kp20k	en	40000	5.26	156.54	63.3
Inspec	en	2000	9.64	124.36	78.5
Krapivin	en	2304	5.336	156.87	56.2
NUS	en	211	11.66	164.80	51.3
SemEval	en	244	15.42	173.77	44.5
DUC	en	308	8.064	683.14	96.6

Recall@$\mathcal{O}$. Recall@$\mathcal{O}$ is the ratio of the number of relevant keywords ranked higher or equal than $\mathcal{O}$ by the system divided by the number of correct ground truth keywords. This measure is sometimes also called R-precision (Zesch & Gurevych, 2009).

$$recall = \frac{|relevant \quad keywords@\mathcal{O}|}{|correct \quad keywords|} \tag{4}$$

F1@$\mathcal{O}$. Harmonic mean between Precision@$\mathcal{O}$ and Recall@$\mathcal{O}$

$$F1@\mathcal{O} = 2 * \frac{P@\mathcal{O} * R@\mathcal{O}}{P@\mathcal{O} + R@\mathcal{O}}$$

Precision@M. Here, M denotes the number of predicted keyphrases. This means no truncation on the predicted keywords is conducted.

$$precision = \frac{|relevant \quad keywords@M|}{|returned \quad keywords|} \tag{5}$$

Recall@M. Recall@M is the ratio of the number of relevant keywords returned by the system divided by the number of correct ground truth keywords.

$$recall = \frac{|relevant \quad keywords@M|}{|correct \quad keywords|} \tag{6}$$

F1@M. Harmonic mean between Precision@M and Recall@M

$$F1@M = 2 * \frac{P@M * R@M}{P@M + R@M}$$

Illustrative examples showing how exactly some of these measures are computed are given in Table 2.

Table 2: Example results for evaluation measures used in our approaches to keyword extraction.

Predicted keywords	Ground truth	p@10	r@10	p@$\mathcal{O}$	r@$\mathcal{O}$	p@M	r@M
k1, k2, k3, k4, k5, k6, k7, k8, k9, k10, k11	k1, k2, k3, k7, k11	4/11	4/5	3/11	3/5	5/11	5/5
k1, k2, k3, k7	k1, k2, k3, k4, k5	3/4	3/5	3/4	3/5	3/4	3/5

In order to compare the results of our approaches to other state-of-the-art approaches, we use the same evaluation methodology as Yuan et al. (2019), F1@k and F1@$\mathcal{O}$, where k is either 5 or 10. Both F1@k and F1@$\mathcal{O}$ are calculated as a harmonic mean of macro-averaged precision and recall, meaning that precision and recall scores for each document are averaged and the F1 score is calculated from these averages. To have comparable results, lowercasing and stemming are performed on both the gold standard keywords and generated keyphrases during evaluation. Only keywords that appear in a text (present keywords) were used as a gold standard in order to make the results of the conducted experiments comparable with the reported results from the related work.

2.2 Related work on terminology extraction and alignment

After presenting related work on keyword extraction, we continue with term extraction methods as term- and keyword extraction methods are highly overlapping. We start by providing a clarification regarding the terminology used in this report. Term extraction, also called automatic term extraction (ATE) or automatic term recognition (ATR) "is the automated process of identifying terms in specialised texts, where terms can be described as the linguistic representations of domain-specific concepts" (Rigouts Terryn et al., 2019). For term extraction, we can distiguish between monolingual and bilingual term extraction. In monolingual setting, only corpora in a single language are considered. For bilingual extraction, following the distinction between two basic approaches made by (Foo, 2012):

- *extract-align* where we first extract monolingual candidate terms from both sides of the corpus and then align the terms, and
- *align-extract* where we first align single and multi-word units in parallel sentences and then extract the relevant terminology from a list of candidate term pairs,

we propose the following two definitions:

- *Bilingual terminology extraction* is the process which, given the input of related specialized monolingual corpora, results in the output of terms aligned between two languages. The process can either start with extracting monolingual candidate terms and aligning them between two languages (i.e. extract-align) or with aligning phrases and then extracting terms (i.e. align-extract) or any other sequence of actions.
- *Bilingual terminology alignment* is the process of aligning terms between two candidate term lists in two languages.

Bilingual terminology alignment has a narrower focus than bilingual terminology extraction, but the two terms are often used interchangeably in various papers.

In this section we present the related work on monolingual and bilingual terminology extraction and alignment in Sections 2.2.1 and 2.2.2, followed by Section 2.2.3 that presents the analysis of past papers on bilingual terminology extraction from the point of view of reproducibility and replicability. The section is based on the published paper with the title: Reproduction, replication, analysis and adaptation of a term alignment approach by Repar, Martinc, & Pollak (2019).

2.2.1 Monolingual terminology extraction methods

We start with a brief overview of the state-of-the-art monolingual term extraction methods. Term extraction "is the automated process of identifying terms in specialised texts, where terms can be described as the linguistic representations of domain-specific concepts" (Rigouts Terryn et al., 2019). In the broadest sense, there are two different approaches to term extraction: linguistic and statistical.

- The linguistic approach utilizes the distinctive linguistic aspects of terms - most often their syntactic patterns.
- The statistical approach takes advantage of term frequencies in the corpus.

Most state-of-the-art systems are hybride, using a combination of the two approaches: e.g., Justeson and Katz (1995) first define part-of-speech patterns of terms and then use simple frequencies to filter the term candidates.

Many terminology extraction algorithms are based on the concepts of termhood and unithood defined by Kageura & Umino (1996). Termhood is "the degree to which a stable lexical unit is related to some domain-specific concepts" and unithood is "the degree of strength or stability of syntagmatic combinations and collocations." Termhood-based statistical measures function on a presumption that a term's relative frequency will be higher in domain-specific corpora than in the general language. Several approaches utilizing termhood have been developed, including those by Khurshid et al. (2000) and Vintar (2010). Common statistical measures are used to measure unithood, such as mutual information (Daille et al., 1994) or t-test (Wermter & Hahn, 2005). Other state-of-the-art models include Termostat (Drouin, 2003) and Termolator (Meyers et al., 2018). In the last few years, word embeddings have become a very popular natural language processing technique. The turning point was the paper by Mikolov et al. (2013) describing word2vec, a word embedding toolkit that can create vector space models much faster than previous attempts. Several attempts have already been made to utilise word embeddings for terminology extraction (e.g., Amjadian et al., 2016; Wang et al., 2016; Gao & Yuan, 2019).

2.2.2 Bilingual terminology extraction methods

The primary purpose of bilingual terminology extraction is to build a term bank, i.e. a list of terms in one language along with their equivalents in the other language. With regard to the input text, we can distinguish between alignment on the basis of a parallel corpus and alignment on the basis of a comparable corpus. For the translation industry, bilingual terminology extraction from parallel corpora is extremely relevant due to the large amounts of sentence-aligned parallel corpora available in the form of translation memories (in the TMX file format). Consequently, initial attempts at bilingual terminology extraction involved parallel input data (Kupiec, 1993; Daille et al., 1994; Gaussier, 1998), and the interest of the community continued until today (Ha et al., 2008; Ideue et al., 2011; Macken et al., 2013; Haque et al., 2014; Arčan et al., 2014; Baisa et al., 2015). However, most parallel corpora are owned by private companies[1], such as language service providers, who consider them to be their intellectual property and are reluctant to share them publicly. For this reason (and in particular for language pairs not involving English) considerable efforts have also been invested into researching bilingual terminology extraction from comparable corpora (Fung & Yee, 1998; Rapp, 1999; Chiao & Zweigenbaum, 2002; Cao & Li, 2002; Daille & Morin, 2005; Morin et al., 2008; Vintar, 2010; Bouamor et al., 2013; Hazem & Morin, 2016, 2017).

Despite the problem of bilingual term alignment lending itself well to the binary classification task, there have been relatively few approaches utilizing machine learning. For example, similar to Aker et al. (2013), Baldwin & Tanaka (2004) generate corpus-based, dictionary-based and translation-based features and train an SVM classifier to rank the translation candidates. Note that they only focus on multi-word noun phrases (noun + noun). A similar approach, again focusing on noun phrases, is also described by Cao & Li (2002). Finally, Nassirudin & Purwarianti (2015) also reimplement Aker et al. (2013) for the Indonesian-Japanese language pair and further expand it with additional statistical features.

2.2.3 Analysis of papers on bilingual terminology extraction from the viewpoint of reproducibility and replicability

In an ideal reproducibility and replicability scenario, a scientific paper would contain an accurate and clear description of the datasets used and experiments conducted and the authors would provide a single link containing all the datasets (versions, subsets etc.) used for the experiments along with the experiment source code (or alternatively, an online tool to run the experiments). These could then be used to replicate the experiments and reproduce the results using the descriptions provided in the paper.

As reported in our paper (Repar, Martinc, & Pollak, 2019), we have analyzed several[2] bilingual terminology extraction papers from the past 25 years from the point of view of dataset, code and tool availability. The summary of results is available in Table 3.

[1] However, some publicly available parallel corpora do exist. A good overview can be found at the OPUS web portal (Tiedemann, 2012).

[2] The selection process was as follows: the starting point were selected seminal papers on the field, as well as two queries in the ACL Anthology database: "term alignment" and "bilingual terminology extraction". We analysed the papers found by these two queries as well as any additional papers mentioned in the related works sections of these papers and the main criterion for including a paper in our analysis was that it primarily deals with bilingual terminology extraction (and not for example latent semantic analysis, such as Bader & Chew (2008)). However, no strict systematic review with inclusion and exclusion criteria was made, as such survey would be beyond the needs of this work.

Table 3: Analysis of bilingual terminology extraction papers from the point of view of reproducibility and replicability.

Paper	Dataset	Code	Tool	Citations
Kupiec (1993)	Links	No	No	333
Daille et al. (1994)	No	No	No	268
Fung & Yee (1998)	Description	No	No	427
Gaussier (1998)	No	No	No	84
Rapp (1999)	Description	No	No	552
Chiao & Zweigenbaum (2002)	Description	No	No	135
Cao & Li (2002)	Description	No	No	141
Morin et al. (2007)	No	No	No	113
Daille & Morin (2005)	Obsolete	No	Obsolete	56
Morin et al. (2008)	Links	No	Obsolete	22
Ha et al. (2008)	Description	No	No	4
Lee et al. (2010)	Description	No	No	22
Vintar (2010)	No	No	Obsolete	53
Ideue et al. (2011)	No	No	Yes [1]	9
Macken et al. (2013)	No	No	No	48
Bouamor et al. (2013)	Description	No	No	24
Aker et al. (2013)	Links	No	No	36
Arčan et al. (2014)	Links	No	No	18
Haque et al. (2014)	Links	No	No	11
Kontonatsios et al. (2014)	Description	No	No	14
Baisa et al. (2015)	No	No	Yes	5
Hazem & Morin (2016)	Links	No	No	12
Hazem & Morin (2017)	Links	No	No	2

[1] A Perl module (Term Extract) was used, however the link leads to a Japanese website.

Dataset availability

In terms of dataset availability, we looked at whether the paper contains some description of how the datasets were constructed and which could (theoretically) be used to reconstruct the datasets. Note that under "dataset", we include corpora, gold standard termlists, seed dictionaries and all other linguistic resources needed to conduct the experiments in the paper. For example, we consider the following paragraph from Rapp (1999) to be a valid description of a dataset: *As the German corpus, we used 135 million words of the newspaper Frankfurter Allgemeine Zeitung (1993 to 1996), and as the English corpus 163 million words of the Guardian (1990 to 1994).* On the other hand, this paragraph from Ideue et al. (2011) is not considered a valid description: *We extracted bilingual term candidates from a Japanese-English parallel corpus consisting of documents related to apparel products.* In the former example, dataset reconstruction would be difficult but not impossible, while in the latter it is impossible. An even better option is to link to actual datasets or refer to papers where datasets are described and linked, which is why we also looked for dataset links and/or references in the analyzed papers. Note that there are several examples where links are provided only for a selection of the datasets used in the experiments e.g., by Morin et al. (2008).

As evident from Table 3, dataset availability is the least problematic aspect of reproducibility and replicability in terminology (extraction and) alignment papers with approximately two thirds of the analysed papers (15 out of 23) either containing a description of the resources used for the experiments, providing links to them or referring to papers where they are described.

We expected the earlier papers to have less information on datasets than latter ones, but this turned out not to be the case. In fact, the earliest paper analyzed by Kupiec (1993) provides a reference to a publicly available corpus (Canadian Hansards: Gale & Church, 1993). The first paper to have a separate

section with data/resource description is (Rapp, 1999) and from this point on, almost all papers have such a section—usually titled “Data and Resources”, “Resources and Experimental Setup”, “Linguistic resources” or similar.

However, it is rarely documented what version of the dataset was used and whether an entire dataset was used or only a part of it (as in random selection, train-test split, etc.). In most cases, little information is provided on the actual subsets used for the experiments. Another aspect of dataset use is the languages: when one of the languages involved is English, it is much easier to find datasets than for other language combinations. Finally, there is also the issue of keeping the links active. For example, many of the links in (Daille & Morin, 2005) and (Morin et al., 2008) are not active anymore while Bouamor et al. (2013) state that the corpora and terminology gold standard lists created for the paper will be shared publicly, but no links are provided.

The most significant problem encountered during our analysis was the fact that terminology alignment is most often not the sole focus of a paper, such as in the paper by Haque et al. (2014), where the experiments start with monolingual terminology extraction from two languages and the extracted terms are then aligned. As terminology extraction and alignment go hand-in-hand, it may often be impossible to make a clear distinction between the terminology extraction and terminology alignment datasets. This means that the dataset results in Table 3 are not a true apple-to-apple comparison: one paper might link to the parallel corpus used to extract terms from, while another to a gold standard termlist. Our main criterion was whether the dataset description (or link) could be used to replicate the experiments described in the paper.

An ideal terminology (extraction and) alignment dataset would therefore consist of a bilingual or multilingual (parallel or comparable) corpus along with reference (gold standard) term lists containing terms that can be found in the corpus. Such corpora are TTC wind energy and TC mobile technology,[3] which contain data for six languages (English, French, German, Spanish, Russian, Latvian, Chinese), or the Bitter corpus,[4] which contains data for the EN-IT language pair. The first was used in (Hazem & Morin, 2016), while the second one by Arčan et al. (2014). Since such datasets are scarce, researchers employ various methodologies for constructing their own datasets. One method, used by Aker et al. (2013), is to take one of the available multilingual translation memories containing EU documentation (such as Europarl (Koehn, 2005) or DGT (Steinberger et al., 2013)) as the corpus and a glossary (e.g., IATE (Johnson & Macphail, 2000)) or thesaurus (e.g., Eurovoc (Steinberger et al., 2002)) as the terminology gold standard list. Another strategy, used by Hazem & Morin (2017), is to collect a comparable corpus manually (i.e. scientific articles in French and English from the Elsevier[5] website) and a domain specific terminological resource (i.e. UMLS[6]) as a reference termlist. Hazem & Morin (2017) also filter out those terms from the termlist that do not appear often enough in their corpus. In other cases (e.g., Haque et al., 2014), the datasets are not available because the papers were written as part of industrial projects and the datasets are private.

Parallel to our research, Rigouts Terryn et al. (2019) have come to similar conclusions and have started building a multilingual terminology extraction dataset according to their best practice recommendations. However, the dataset will be available only at the end of their project.

Code and tool availability

We have discovered that no paper has made experiment code available and only a few provide access or links to tools where the experiments were conducted. But even when links to tools are provided, reproducibility and replicability may be hindered: for example, the link provided in (Ideue et al., 2011) leads to a Japanese website. Another issue is the long-term availability of resources. For example,

[3] http://www.lina.univ-nantes.fr/?Reference-Term-Lists-of-TTC.html
[4] https://hlt-mt.fbk.eu/technologies/bittercorpus
[5] https://www.elsevier.com/
[6] https://www.nlm.nih.gov/research/umls/

Daille & Morin (2005) conducted their experiments in *ACABIT*, an open source terminology extraction software. However, the link given in the paper does not work anymore. From the analyzed papers, the only example of bilingual term extraction and alignment tool, which is publicly available, is the Sketch Engine term extraction module, described by Baisa et al. (2015).

None of the papers analysed in this section fulfill the ideal scenario described at the start of this section (i.e. a single link with code and all datasets) which severely hinders any replicability attempts as will be evident from our own experiments described in this paper.

3 Keyword extraction

This section presents our work in the area of keyword extraction. Section 3.1 presents our attempts to reproduce the results of the unsupervised state-of-the-art keyword extraction method. Next, we introduce novel methods for keyword extraction: a graph-based unsupervised method is presented in Section 3.2, while Section 3.3 reports on our promising supervised approach called TNT-KID, a transformer-based neural tagger for keyword identification. We compare the results of our and state-of-the-art supervised and unsupervised approaches in Section 3.4. The final experiments related to keyword extraction consist of testing selected approaches on Croatian EMBEDDIA datasets in Section 3.5.

3.1 Attempts in reproducing YAKE results

As YAKE (Campos et al., 2018a,b) reports to have one of the best results for unsupervised keyword extraction, we attempted to reproduce their published results. We discuss in detail our attempts at reproducing the results of YAKE.

In our initial experiments, we first attempted to reproduce YAKE's results by simply installing it via the pke library[7], followed by computation of keyword matches, where the detected keywords as well as the gold standard ones were first stemmed. From the measures described in Section 2.1.4, f1@10 was reported. The pke implementation's default settings did not achieve competitive performance (as reported in `https://github.com/LIAAD/yake`). We next discuss in detail our attempts and consolidation.

As we were not able to reproduce the results using the package manager version (pke library), we attempted to reproduce the results using the YAKE's official repository. We initially consulted (Campos et al., 2018a) and (Campos et al., 2018b), which was updated with the newest version that reflects the upcoming January 2020 full paper (Campos et al., 2020). At this point, we still observed discrepancies across datasets (even though we used the same default setting as reported in the paper), however, the official, most recent version of YAKE performed notably better than the previously published one in the pke library.

We considered several possible causes of discrepancies, including higher k (f1@20), f1@M instead of f1@10, stemming or not, only keywords present in the document considered for the gold standard etc., however testing each of these hypotheses did not help with reproducing the authors' published results.

In the following round of experiments, we consulted the YAKE authors, who offered us their Python script, which, along with the TREC (Text Retrieval Conference) evaluation scripts supposedly reproduces the results[8]. However, even when using this script and the TREC evaluation, the results remained very similar to the ones obtained in the previous experiments. Our attempt is accessible online[9].

[7] `https://github.com/boudinfl/pke`

[8] `https://github.com/vitordouzi/ConvertKeyphrase2TREC` and `https://trec.nist.gov/trec_eval/`

[9] `https://github.com/SkBlaz/rakun/tree/master/reproduce_yake`

At this point, we decided to re-evaluate YAKE, as available on their official GitHub repository against the state-of-the-art baselines in our evaluation environment, using newly introduced evaluation code as part of the most recent neural sequence-to-sequence approach (Yuan et al., 2019). The results are reported in Table 4 in Section 3.4.

3.2 Novel unsupervised approach to keyword extraction: RaKUn

We next summarise our keyword extraction approach RaKUn, published in a conference paper by Škrlj et al. (2019). RaKUn operates in three main steps:

1. transformation of texts into a graph,
2. graph pruning, and
3. token ranking (keyword detection).

The key novelty of the RaKUn algorithm is the capability to aggregate tokens based on their similarity, potentially decreasing redundancy of the token space. Further, RaKUn employs *load centrality*, as a fast measure of vertex centrality that captures the importance of keywords comprised of a single, two or three tokens.

3.2.1 Representing text

In this work we consider *directed* graphs. Let $G = (V, E)$ represent a graph comprised of a set of vertices V and a set of edges ($E \subseteq V \times V$), which are ordered pairs. Further, each edge can have a real-valued weight assigned. Let $\mathcal{D}$ represent a document comprised of tokens $\{t_1, \ldots, t_n\}$. The order in which tokens in text appear is known, thus $\mathcal{D}$ is a totally ordered set.

A potential way of constructing a graph from a document is by simply observing word co-occurrences. When two words co-occur, they are used as an edge. However, such approaches do not take into account the sequential nature of words in a document, meaning that the order is lost. We attempt to take this aspect into account as follows.

The given corpus is traversed, and for each element t_i, its successor t_{i+1}, together with a given element, forms a directed edge $(t_i, t_{i+1}) \in E$. Finally, such edges are weighted according to the number of times they appear in a given corpus. Thus the graph, constructed after traversing a given corpus, consists of all local neighborhoods (order one), merged into a single joint structure. Global contextual information is potentially kept intact (via weights), even though it needs to be detected via network analysis as proposed next.

3.2.2 Improving graph quality by meta vertex construction

A naïve approach to constructing a graph, as discussed in the previous section, commonly yields noisy graphs, rendering learning tasks harder. Therefore, we next discuss the selected approaches we employ in order to reduce both the computational complexity and the spatial complexity of constructing the graph, as well as increasing its quality (for the given down-stream task).

First, we consider the following heuristics that can reduce the complexity of the graph constructed for keyword extraction: token length (while traversing document $\mathcal{D}$, only tokens of length $\mu > \mu_{\text{min}}$ are considered), and lemmatization (tokens can be lemmatized, offering spatial benefits and avoiding redundant vertices in the final graph). The two modifications yield a potentially simpler graph, which is more suitable and faster for mining.

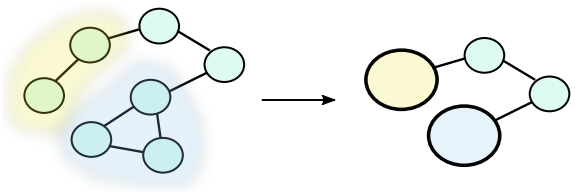

Figure 1: Meta vertex construction. Sets of highlighted vertices are merged into a single vertex. The resulting graph has less vertices, as well as edges.

Even if the optional lemmatization step is applied, one can still aim at further reducing the graph complexity by merging similar vertices. This step is called *meta vertex construction*. The motivation can be explained by the fact, that even similar lemmas can be mapped to the same keyword (e.g., mechanic and mechanical; normal and abnormal). This step also captures spelling errors (similar vertices that will not be handled by lemmatization), spelling differences (e.g., British vs. American English), non-standard writing (e.g., in Twitter data), mistakes in lemmatization or unavailable or omitted lemmatization step.

The meta-vertex construction step works as follows. Let V represent the set of vertices, as defined above. A meta vertex M is comprised of a set of vertices that are elements of V, i.e. $M \subseteq V$. Let M_i denote the i-th meta vertex. We construct a given M_i so that for each $u \in M_i$, u's initial edges (prior to merging it into a meta vertex) are rewired to the newly added M_i. Note that such edges connect to vertices which are not a part of M_i. Thus, both the number of vertices, as well as edges get reduced substantially. This feature is implemented via the following procedure:

1. Meta vertex candidate identification. Edit distance and word length distance[10] are used to determine whether two words should be merged into a meta vertex (only if length distance threshold is met, the more expensive edit distance is computed).
2. The meta vertex creation. As common identifiers, we use the stemmed version of the original vertices and if there is more than one resulting stem, we select the vertex from the identified candidates that has the highest centrality value in the graph and its stemmed version is introduced as a novel vertex (meta vertex).
3. The edges of the words entailed in the meta vertex are next rewired to the meta vertex.
4. The two original words are removed from the graph.
5. The procedure is repeated for all candidate pairs.

A schematic representation of meta vertex construction is shown in Figure 1. The yellow and blue groups of vertices both form a meta vertex, the resulting (right) graph is thus substantially reduced, both with respect to the number of vertices, as well as the number of edges.

3.2.3 Keyword identification

After previous steps, where a graph is constructed from a given ordered set of tokens and merging of very similar vertices by meta-vertex construction step, in this smaller, denser graph, load centrality is computed for each vertex. Note that at this point, should the top k vertices by centrality be considered, only single term keywords emerge. The method is extended to cover 2- and 3-grams. Having obtained

[10]The edit distance (Levenshtein) measures the similarity of two character sequences by transforming the first one into the second one using a dynamic programming paradigm. The word length distance is the difference of lengths between two words.

a set of (keyword, score) pairs, we finally sort the set according to the scores (descendingly), and take top k keywords as the result.

In this report, the RaKUn keyword extraction is evaluated against the the most recent version of YAKE (Campos et al., 2020) (see Section 3.4), which was not available at the time of writing of the original paper, as well as to the other state-of-the art approaches on multiple datasets.

In future work we will investigate how embeddings-based measures can be used in vertex aggregation step.

3.3 Novel supervised approach to keyword extraction: TNT-KID

In this section we describe TNT-KID: Transformer-based Neural Tagger for Keyword IDentification.

The unsupervised approaches, such as RaKUn (Škrlj et al., 2019) and YAKE (Campos et al., 2018b), have many advantages over supervised approaches for keyword extraction (they are language and genre independent, do not require any training and are computationally undemanding) but also a couple of crucial deficiencies:

- TF-IDF and graph based features such as PageRank, used by these systems to detect the importance of each word in the document, are based only on simple statistics like word occurrence and co-occurrence, and are therefore unable to grasp the entire semantic information of the text.
- Since these systems cannot be trained, they can not be adapted to the specifics of the syntax, semantics, content, genre and keyword tagging regime of a specific text. On the other hand, supervised approaches have direct access to the gold standard keyword set for each text during the training phase, enabling more efficient adaptation.

These deficiencies result in a much worse performance when compared to the state-of-the-art supervised algorithms (see Table 4). Therefore, besides developing a state-of-the-art unsupervised keyword extractor RaKUn in the scope of the EMBEDDIA project, we are also currently developing a neural supervised keyword extractor capable of overcoming the aforementioned deficiencies. In order to successfully grasp the semantic information of the text, we propose a transfer learning technique, where a classifier is first trained as a language model on a large corpus and then fine-tuned on a (usually) small-sized corpus with manually labeled keywords.

Unlike other proposed neural keyword extractors (Meng et al., 2017, 2019; Yuan et al., 2019), we do not employ recurrent neural networks but instead opt for a transformer architecture (Vaswani et al., 2017). Secondly, while these approaches formulate a keyword extraction task as a sequence-to-sequence generation task, where the classifier is thought to generate an output sequence of tokens step by step according to the input sequence and the previous generated output tokens, we formulate a keyword extraction task as a sequence labeling task, similar as Gollapalli et al. (2017) and Luan et al. (2017).

The system is currently still in development, but we can already show that our system offers only slightly worse performance than state-of-the-art sequence-to-sequence systems on test sets containing abstracts of scientific papers with long documents but is on the other hand also capable of drastically outperforming these systems on corpora with shorter average document length and more keywords per document, which is of special interest in EMBEDDIA news media applications. We also show that, due to transfer learning, our system does not require large manually labeled training corpora required by sequence-to-sequence models, which makes it transferable to low resourced languages where such datasets are not available.

3.3.1 Approach

Our approach relies on a transfer learning technique (Howard & Ruder, 2018; Devlin et al., 2018), where a neural model is first pretrained as a language model on a large corpora. This model is then fine-tuned for each specific keyword detection task on each specific manually labeled corpus by adding and training the final keyword labeling layer. With this approach, the syntactic and semantic knowledge of the pretrained language model is transferred and leveraged in the keyword detection task, allowing for good performance of the keyword detection model even on small datasets.

The model follows an architectural design of a transformer (Vaswani et al., 2017), more specifically the GPT-2 language model (Radford et al., 2019) with a couple of significant modifications:

- The standard input embedding layer and softmax function were replaced by adaptive input representations (Baevski & Auli, 2018) and an adaptive softmax (Grave et al., 2017). These modifications drastically reduce the memory requirements and time complexity of the original model at the expense of a marginal drop in performance.
- Absolute positional embeddings used in the original model were replaced by relative positional embeddings, as in Dai et al. (2019). The main idea is to only encode the relative positional information in the hidden states instead of the absolute. This approach slightly improves the performance of the model and also requires a re-parameterization of the attention mechanism.
- Besides the text input, the model takes an additional part-of-speech (POS) tag sequence as an input. This sequence is first embedded and then added to the word embedding matrix.

During pretraining, the model is trained as a standard language model, where the task can be formally defined as predicting a probability distribution of words from the fixed size vocabulary V, for word w_{t+1}, given the historical sequence $w_{1:t} = [w_1, ..., w_t]$.

During fine-tuning, the final densely connected layer of the language model is replaced with a new dense layer of size $SL * NC$, where SL stands for sequence length (i.e. number of words in the input text) and NC stands for number of classes. Since each word in the sequence can either be a keyword (or at least part of the keyphrase) or not, the keyword tagging task can be modeled as a binary classification task, where the model is trained to predict if a word in the sequence is a keyword or not. Figure 2 shows an example of how an input text is first transformed into a numerical sequence that is used as an input of the model, which is then trained to produce a sequence of zeroes and ones, where the positions of ones indicate the positions of keywords in the input text. Negative Log Loss function between the correct sequence of zeroes and ones and the predicted sequence is used during training. Since a large majority of words in the sequence are not keywords, we assign a larger weight to the positive class in order to prevent the majority negative class to prevail.

In order to produce final set of keywords for each document, tagged words are extracted from the text and duplicates are removed. Note that a sequence of ones is always interpreted as a multi-word keyphrase and not a combination of one-worded keywords (e.g., *distributed interactions* from Figure 2 is considered as a single multi-word keyphrase and not as two distinct one word keywords). After that, the following filtering is conducted:

- If a keyphrase is longer than four words, it is discarded.
- Punctuation (with the exception of dashes and apostrophes) is removed from keywords.
- The detected keyphrases are ranked and arranged according to the softmax probability assigned by the model in a descending order.
- Lowercasing and stemming are performed on both the gold standard keywords and generated keyphrases during evaluation, as in related work.

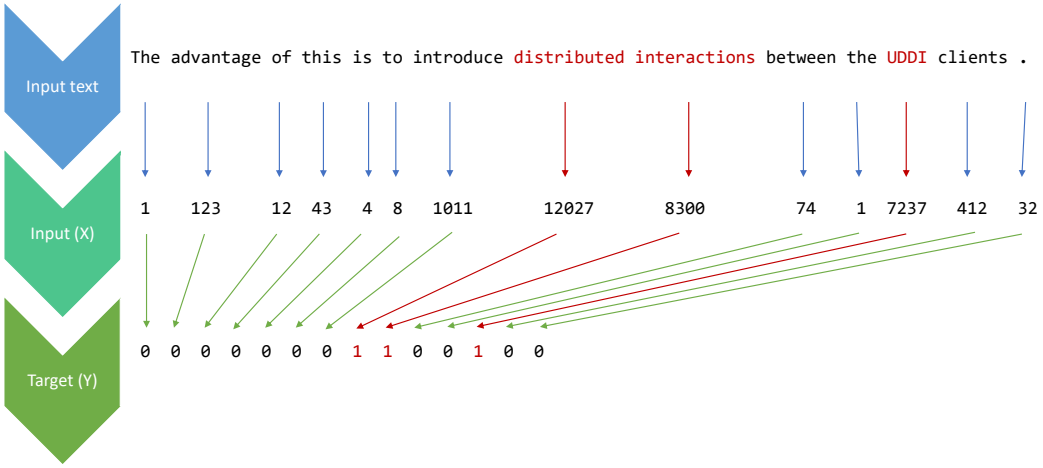

Figure 2: Encoding of the input text "*The advantage of this is to include distributed interactions between the UDDI clients.*" with keywords *distributed interactions* and *UDDI*. In the first step, the text is converted into a numerical sequence, which is used as an input to the model. The model is trained to convert this numerical sequence into a sequence of zeroes and ones, where the ones indicate the position of a keyword.

3.3.2 Experimental setting

We conducted experiments on the datasets described in Section 2.1.3. First, we trained a language model on a concatenation of texts from all the datasets. After that, the trained language model was fine-tuned for each dataset on a train set and tested on a test set. A validation set (encompassing random 20% of documents from a train set) was used in order to determine the best hyperparameters of the model. The model was fine-tuned for a maximum of 10 epochs and after each epoch the trained model was tested on the validation set. The model with the best performance on the validation set (in terms of Negative Log Loss) was used for keyword detection on the test set. All combinations of the following hyperparameter values were tested before choosing the best combination, which is written in bold in the list below:

- Learning rates: 0.0001, 0.00005, **0.00001**, 0.00003
- Embedding size: 256, **512**
- Number of attention heads: 4, **8**, 12
- Sequence size: 128, **256**
- Number of attention layers: 4, **8**, 12
- Weight for a positive class: 1, 2, 4, **6**, 8

The same train-test splits are used as in related work Meng et al. (2017), for a fair comparison of the models.

Table 4: Empirical evaluation of state-of-the-art keyword extractors.

	Results reported by original authors							Implemented or reimplemented by us		
	TfIdf	TextRank	KEA	Maui	CopyRNN	CopyRNN-improved	CatSeqD	YAKE	RaKUn	TNT-KID
	Kp20k									
F1@5	0.072	0.181	0.046	0.005	0.328	0.317	**0.348**	0.134	0.177	0.285
F1@10	0.094	0.151	0.044	0.005	0.255	0.273	**0.298**	0.136	0.16	0.263
F1@$\mathcal{O}$	0.063	0.184	0.051	0.004	*	0.335	**0.357**	0.116	0.16	0.275
	Inspec									
F1@5	0.160	0.286	0.022	0.035	0.292	0.244	0.276	0.202	0.101	**0.449**
F1@10	0.244	0.339	0.022	0.046	0.336	0.289	0.333	0.222	0.108	**0.505**
F1@$\mathcal{O}$	0.208	0.335	0.022	0.039	*	0.290	0.331	0.212	0.108	**0.491**
	Krapivin									
F1@5	0.067	0.185	0.018	0.005	0.302	0.305	**0.325**	0.204	0.127	0.271
F1@10	0.093	0.160	0.017	0.007	0.252	0.266	**0.285**	0.186	0.106	0.259
F1@$\mathcal{O}$	0.068	0.211	0.017	0.006	*	0.325	**0.371**	0.170	0.106	0.240
	NUS									
F1@5	0.112	0.230	0.073	0.004	0.342	**0.376**	0.374	0.130	0.224	0.279
F1@10	0.140	0.216	0.071	0.006	0.317	0.352	**0.366**	0.186	0.193	0.286
F1@$\mathcal{O}$	0.122	0.238	0.081	0.006	*	**0.406**	**0.406**	0.154	0.193	0.271
	SemEval									
F1@5	0.088	0.217	0.068	0.011	0.291	0.318	**0.327**	0.151	0.167	0.282
F1@10	0.147	0.226	0.065	0.014	0.296	0.318	**0.352**	0.212	0.159	0.309
F1@$\mathcal{O}$	0.113	0.229	0.066	0.011	*	0.317	**0.357**	0.168	0.159	0.278
	DUC									
F1@5	0.101	*	*	*	*	0.083	*	0.122	0.189	**0.302**
F1@10	0.120	0.097	*	*	0.165	0.107	*	0.148	0.172	**0.336**
F1@$\mathcal{O}$	0.115	*	*	*	*	*	*	0.135	0.172	**0.336**

3.4 Keyword extraction results compared to the state-of-the-art

In Table 4, we present the results achieved by a number of algorithms on the datasets presented in Table 1. Evaluation measures were presented in Section 2.1.4. Only keywords which appear in a text (present keywords) were used as a gold standard in order to make the results of the conducted experiments comparable with reported results from the related work. One more issue requiring consideration is the difference in training regimes. TfIdf, TextRank, YAKE and RaKUn algorithms are unsupervised and do not require any training, KEA, Maui and TNT-KID were trained on a different train set for each of the datasets, and CopyRNN, CopyRNN-improved and CatSeqD were all trained on a large Kp20K dataset, since they require a large train set for competitive performance.

First, we comment on the results comparing RaKUn (Škrlj et al., 2019) to YAKE (Campos et al., 2020) as reported to be the best performing unsupervised system. The purpose of this comparison is to demonstrate the objective performance of both the RaKUn algorithm developed in EMBEDDIA, as well as to re-evaluate YAKE's performance in an end-to-end manner, using the same evaluation as for the other approaches. We intentionally report default hyperparameter settings, as both we, the authors of RaKUn, as well as YAKE's authors claim that a single hyperparameter set can offer sufficient performance across multiple datasets.

The empirical evaluation confirms our reproducibility findings that YAKE by default does not perform as well as claimed, even though we did not test extensive hyperparameter combinations. In fact TextRank appears to be the best unsupervised algorithm, while RaKUn performs on-par with other state-of-the-art methods, including YAKE.

Overall, supervised neural network approaches drastically outperform all other approaches. Among them, our proposed Transformer-based Neural Tagger for Keyword IDentification (TNT-KID) manages to outperform state-of-the-art approaches on two out of four datasets by a large margin.

TNT-KID manages to outperform the CatSeqD approach by a margin of about 17 percentage points according to all criteria on the Inspec dataset. On the DUC dataset, it again outperforms the second best neural approach (CopyRNN) by about 17 percentage points according to the F1@10 score. On other datasets, it generally achieves about 3-4 percentage points worse results than the best reported

approach according to the F1@10 measure, with the exception of the NUS dataset, where the difference is about 8 percentage points. When it comes to the F1@5 and F1@$\mathcal{O}$ measures, the differences between TNT-KID and the best performing system vary more, ranging from about 13 percentage points difference in F1@$\mathcal{O}$ on the NUS dataset, to the about 5 percentage points difference in F1@5 on the SemEval an Krapivin datasets.

This can mostly be explained by the differences in how the two systems work and the average number of gold standard keywords per document. On average, the CatSeqD system detects less than 5 keywords per document, since it was trained on the large Kp20K dataset, where the average document has just 3.33 (63.3% out of 5.26 keywords per document) keywords that appear in the text. This means that this system will in most cases maximize precision and neglect recall. On the other hand, our system is trained to maximize recall and on average predicts about 10 keywords per document, which off course also hurts precision of the system, especially at smaller k values and at smaller $\mathcal{O}$ values (this happens on datasets with not many keywords present in the documents). For this reason, our system manages to outperform other state-of-the-art system on Inspec (about 7.5 present keywords per document) and DUC (about 8 present keywords per document). Inspec is also a dataset with on average shortest documents (on average 124.36 words long), which might negatively influence the performance of CopyRNN and CatSeqD, that on average produce about one keyword per 50 words. While we do not have any reported score for CatSeqD on the DUC dataset, the CopyRNN and CopyRNN-improved perform very poorly on this dataset and are being outperformed even by the unsupervised RaKUn algorithm. The reason for this is mostly likely that the DUC dataset contains news articles and both of these networks were trained on scientific articles.

The difference between TNT-KID and other neural nets in training and prediction regimes implies that the choice of a network is somewhat dependent on the use-case. If a large training dataset of an appropriate genre with manually labeled keywords is available and if the system does not need to predict many keywords, than CatSeqD is most likely the best choice. On the other hand, if only a small train set is available and it is preferable to predict larger number of keywords, than TNT-KID is most likely a better choice. We are still working on further improvement of the system on which we will report in the deliverable due in M24.

3.5 Keyword extraction from the Croatian news dataset of Styria

Styria Media Group is one of the leading media groups in Austria, Croatia, and Slovenia. They publish magazines, daily and weekly newspapers, operate radio stations, TV station and several book publishing companies. The group also operates successful news portals, marketplaces, as well as content and community portals in digital format. Leading portals in the Croatian market in terms of page visits and business results are *24sata* and *Večernji list* which are both managed by Styria.

Styria test dataset, on which we test two keyword extraction algorithms, contains news articles in Croatian from digital editions of *24sata*, *Večernji list*, and their respective niche portals. Each news article contains title, text and on average 3.3 corresponding keywords. Statistics for the Styria dataset are given in Table 5.

Table 5: Statistics of the Styria dataset.

Statistic	Value
No. words	40,512,198
No. documents	142,146
Average no. words per document	285
No. keywords	465,671
Average no. keywords per document	3.276
% Present keywords	57.92

On a given Styria dataset we tested one supervised and one unsupervised approach to keyword extraction. First, we tested the supervised model CopyRNN proposed by Meng et al. (2017), which employs a generative model for keyword prediction with a copying mechanism based on the positional information. We chose this model due to its good performance on the English datasets (see Table 4) and since the code for the model is publicly available on Github (`https://github.com/memray/seq2seq-keyphrase-pytorch`). The model was trained on 600,000 Croatian news articles from *24sata* and *Večernji list* portals. Next, we tested the unsupervised model RaKUn (Škrlj et al., 2019), a language-agnostic graph-based method that requires no training and was implemented in the scope of the EMBEDDIA project.

Based on the evaluation of two tested methods (see Table 6), the CopyRNN model works better then RaKUn according to all criteria. This coincides with the results on the English data, where the supervised models also generally work much better than the unsupervised models. We also notice that the difference in performance between the two models is bigger when we consider only first 5 keywords instead of 10. This is due to the fact that the RaKUn method always returns 10 keywords, while CopyRNN on average returns only 8.34 keywords per documents. A large discrepancy in results in favor of CopyRNN was expected, as CopyRNN is an example of supervised keyword extraction methods that outperform unsupervised methods such as RaKUn. In future we will test also the TNT-KID approach.

Table 6: Evaluation of CopyRNN and RaKUn models on the Styria dataset.

Model	precision@5	recall@5	F1@5	precision@10	recall@10	F1@10	precision@$\mathcal{O}$	recall@$\mathcal{O}$	F1@$\mathcal{O}$
CopyRNN	0.231	0.347	0.28	0.159	0.395	0.23	0.297	0.292	0.29
RaKUn	0.072	0.112	0.09	0.072	0.112	0.09	0.085	0.082	0.08

4 Terminology extraction and alignment

This section presents our work in bilingual terminology extraction and alignment. In Section 4.1 we present a study in which we reimplemented and adapted an approach to term alignment. Next, Section 4.2 reports on TermEnsembler, an ensemble learning approach to bilingual term extraction and alignment. Section 4.3 reports on our experiments with graph-based term alignment using co-frequency information from a bilingual parallel corpus.

4.1 Reimplementation and adaptation of bilingual terminology alignment approach by Aker et al.

This section summarises the reimplementation study that we published in the journal *Language Resources and Evaluation* (Repar, Martinc, & Pollak, 2019), presented in Appendix B.

Our attempts focused on the approach to bilingual term alignment using machine learning by Aker et al. (2013), who consider term alignment as a bilingual classification task: for each term pair, they create various features based on word dictionaries (i.e. created with Giza++ from the DGT translation memory) and word similarities across languages. They evaluated their classifier on a held-out set of term pairs and additionally by manual evaluation. Their results on the held-out set were excellent, with 100% precision and 66% recall for the English-Slovenian and English-French language pair and 98% precision and 82% recall for English-Dutch. The method was therefore selected as being reported as a very strong baseline for aligning lists of terms in many European languages, and could be used for aligning keywords across news datasets in different languages.

Our reproduction attempt focused on three language pairs: English-Slovenian, English-Dutch and English-French (in contrast with the original article where they had altogether 20 language pairs) and we were

unable to reproduce the results following the procedures described in the paper on Eurovoc terms. In fact, our results have been dramatically different from the original paper with precision being less than 4% and recall close to 90% for all three language pairs under consideration. We then tested several different strategies for improving the results ranging from Giza++ dictionary cleaning, lemmatization, different ratios of positive and negative examples in the training and test sets, training set filtering based on feature values and term length, and adding new cognate-based features. The most effective strategies employed unbalanced training set and training set filtering based on certain feature values which resulted in precision exceeding 90% for all three language combinations. It is possible that in the original experiments the authors performed a similar training set filtering strategy, because the original paper mentions that their classifier initially achieved low precision on Lithuanian language training set, which they were able to improve by manually removing positive term pairs that had the same characteristics as negative examples from the training set. However, no manual removal is mentioned for Slovenian, Dutch or French. Further attempts were directed at boosting recall and the performance of cognate-based features. By adding additional cognate-based features, we were able to improve recall by around 16% for Dutch, 8% for French and by around 2% for Slovenian at a cost of a moderate drop in precision.

We also performed manual evaluation similar to the original paper and reached roughly the same results with our adapted approach. In addition, because we discovered that Eurovoc data is of limited use for evaluating the performance of cognate-based features, we ran experiments on an English-Slovenian karstology gold standard term list. With the *Cognates approach* configuration, we improved recall by 11% (compared to the *Training set filtering 3* configuration) and a qualitative analysis of the results showed that the new strategies for boosting the performance of cognate-based features do indeed result in more cognate term pairs being properly aligned. For more details see Appendix B.

4.2 TermEnsembler: An ensemble learning approach to bilingual term extraction and alignment

We developed an ensemble learning approach to bilingual term extraction and alignment, called TermEnsembler (Repar, Podpečan, et al., 2019), described in the paper attached in Appendix C. The system uses a version of the monolingual term extraction approach developed by Pollak et al. (2012), adapted with nested term filtering and near duplicate recognition, to extract terms from English and Slovenian, and then implements seven (existing and novel) approaches to term alignment to align the terms between the two languages (while some of the methods are designed for parallel corpora, other perform also on comparable corpora). The results of the seven methods are then merged together using an evolutionary algorithm in a novel ensemble learning approach. Evaluation of the system showed results with more than 96% of the top 400 term pair alignments produced by the system evaluated as correct by a human evaluator. Even if some of the methods presented imply the use of parallel corpora, which are currently not foreseen in the scope of EMBEDDIA, the methods for alignment from parallel corpora are good baselines for evaluating the cross-lingual embedding techniques that are being developed in WP1.

4.3 Experiment in building a graph-based term alignment approach

We experimented with graph-based approaches for cross-lingual term alignment using co-frequency information from a bilingual parallel corpus. Starting with a list of English and Slovenian terms along with information on individual frequency and co-frequency, we wanted to align the terms in the two lists across two languages. In the first step, we constructed a bipartite network consisting of Slovenian words in one partition and English words in the other partition. The only connections in the network were those linking Slovenian words to co-aligned (i.e. appearing in the aligned sentence pair) English words. The strength of the connection was the number of times the two words appeared in aligned sentences. In the

second step, we ran the Personalized PageRank (PPR, Bahmani et al., 2010) algorithm for each word in the network, producing one feature vector for each word. In the third step, the constructed vectors were used to discover the k nearest words. This was done in one of two ways:

- Using a direct approach, we interpret the PPR vectors as measures of similarity. For each word w, its PageRank vector represents its similarity to other words. Thus, to return k nearest words, we must find the k words for which the corresponding value of w's PageRank has the highest value
- Indirectly, the PPR vectors can be seen as feature vectors representing the words. To return the k nearest words, we therefore find the k words that have the most similar PPR vectors according to some distance metric. In our case, we used the standard Euclidean metric.

However, initial results were not promising and we are now exploring alternative avenues.

4.4 Embedding-based terminology expansion experiments

We developed an approach to extracting domain knowledge from specialized corpora (Pollak et al., 2019), see Appendix D for more details. The technologies related to EMBEDDIA include improving existing statistical terminology extraction, and especially the methodology for terminology expansion with the help of word embeddings, which could be used also for querying archives of news corpora (for expanding manually defined keywords). Before further describing the term expansion experiment, let us recall the definition of *term expansion* from Section 1: "Term expansion is the process of extending a list of existing terms by novel term candidates."

The extraction of specialized knowledge was conducted on a corpus of karstology, i.e. an interdisciplinary domain at the intersection of geology, hydrology, and speleology. The domain is of high interest, as karst is possibly the most prominent geographical feature of Slovenia (with karst formations being some of popular tourist and natural attractions in Slovenia). It is also an interesting example of how terminology is dynamically evolving in a cross-linguistic context. The literature published in English contains many local Slovenian scientific terms and toponyms for typical geomorphological karst structures, which makes it appropriate for research and identification of cognates, as well as homonym terms, with possible differences in meaning across cultures.

The corpus of karstology consists of Slovene, Croatian and English texts. We focus on the Slovene and English parts of the TermFrame corpus (v1.0). The English subcorpus contains about 1.6M words and the Slovene one cca. 1M words (see Table 7 for details).

Table 7: Statistics for English and Slovenian subcorpora.

	English	Slovene
Vocabulary size	64,079	73,813
Documents	24	60
Sentences	103,322	57,575
Words	1,673,132	1,041,475
Tokens	1,972,320	1,231,039
Type-to-token ratio	0.032	0.060

In addition, we are using a short gold standard list of Karst domain terms, called QUIKK termbase[11]. The QUIKK term base consists of terms in four languages, but for the purposes of our experiments, the Slovene and English term lists are used, containing 57 and 185 terms, respectively.

Word embeddings capture certain degree of semantics, as words that are similar or semantically related are closer together in the vector space. Previous research conducted by Diaz et al. (2016) showed that embeddings can be successfully used for expanding queries on topic specific texts. In this research, we

[11] http://islovar.ff.uni-lj.si/karst

test if word embeddings can be used for a similar task of extending the gold standard term lists to find more domain terms. According to the research conducted by Diaz et al. (2016), embeddings trained only on small topic specific corpora outperform non-topic specific general embeddings trained on very large general corpora for the task of query expansion due to strong language use variation in specialized corpora. Therefore, we use the same approach for extending the term list and train custom embeddings on the specialized corpus instead of using pretrained embeddings.

In our experiments, we trained FastText embeddings (Bojanowski et al., 2017) on the Slovenian and English karst subcorpora and use them to find twenty closest words (according to cosine distance between embeddings) for the first fifty terms in the QUIKK term base[12]. These related words are sorted according to their proximity to the term and the first, second, tenth and twentieth ranked words are used in manual evaluation. Embeddings for multi-word terms are generated by averaging the word embeddings for each word in the term.[13]

The method was tested on 47 English and 50 Slovene source terms (i.e. the terms from the gold standard list), for which out of 20 most related words (according to the cosine distance between the source term and the related word), four per each source term were selected for evaluation (first, second, tenth and twentieth ranked words), resulting in 200 term-word pairs for English and 188 for Slovene.[14] Examples of ranked related words for five English and five Slovene terms are presented in Table 8.

Table 8: Examples of ranked related words for five English (upper five examples) and five Slovene (lower five examples) terms.

Term	R1	R2	R10	R20
sinkhole	shakehole	suburban	sinkpoint	dump
aggressive water	aggressively	aggressiveness	qc	coldwater
epikarst zone	epikarstic	subcutaneous	cutaneous	epiphreatic
caprock sinkhole	sinkpoint	overbank	suburb	evacuation
seacave	seacoast	sealevel	vrulja	caveand
udornica	udornina	zapornica	koliševka	kamojstrnik
agresivna voda	sposoben	mehurček	skoznjo	preniči
epikras	epikraški	prenikujoč	epr	vadozen
vrtača	vrtačast	mikrovrtača	globel	neizravnan
rečna jama	reža	narečen	mohoričev	vodokazen

Two human evaluators evaluated the related words according to two criteria:

- Is the word a term
- Semantic similarity to the term

The first criterion is measured on a scale with three nominal classes (*term, karst term, not term*), while the second criterion uses a numerical scale from zero to ten, following the evaluation procedure of Finkelstein et al. (2002), where zero suggests no semantic similarity and ten suggests very close semantic relation (fractional scores were also allowed). The inter-annotator agreement between two evaluators (according to the Cohen's kappa coefficient) is 0.689 for the first criterion and 0.513 for the second criterion for English and 0.594 for the first criterion and 0.389 for the second criterion for the Slovene evaluation.

Table 9 presents embeddings-based term extension results. Out of 200 English term-word pairs, 112

[12] To be exact, 50 English terms, and 47 Slovene terms, since only 47 Slovenian terms from the QUIKK term base appear in the Slovenian corpus.

[13] There are several possible multi-word term aggregation approaches, such as summation of component word vectors, averaging of component word vectors, creating multi-word term vectors, etc. As comparing different techniques is beyond the scope of this study, we decided for the simple averaging technique, as previous research on this topic conducted on the medical domain (Henry et al., 2018) found no statistically significant difference between any multi-word term aggregation method.

[14] In this section, we intentionally name related words as words and not as terms, to contrast them to the gold standard list of terms to which they are compared. As shown in the evaluation, they can be in next step evaluated as terms or not.

Table 9: English and Slovenian embeddings evaluation. *Avg. sem. score* stands for the average of manually prescribed semantic similarity scores for each term-word pair, *Avg. cos. dist* stands for the average cosine distance, *Pearson corr.* is a Pearson correlation coefficient between the semantic similarity score and cosine distance values and *Spearman corr.* is a Spearman correlation coefficient between the semantic similarity score and cosine distance values.

	English				Slovene			
All words	200				188			
Avg. sem. score	3.325				3.859			
Avg. cos. dist.	0.747				0.760			
Pearson corr.	0.181				0.231			
Spearman corr.	0.136				0.194			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	50	50	50	50	47	47	47	47
Avg. sem. score	4.040	3.540	3.110	2.610	4.872	4.468	3.032	3.064
Terms	112				69			
Avg. sem. score	4.710				5.536			
Avg. cos. dist.	0.757				0.771			
Pearson corr.	0.176				-0.018			
Spearman corr.	0.160				-0.016			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	32	30	29	21	17	22	15	15
Karst terms	52				36			
Avg. sem. score	5.702				6.722			
Avg. cos. dist.	0.761				0.780			
Pearson corr.	0.151				-0.152			
Spearman corr.	0.070				-0.067			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	16	14	15	7	12	12	5	7
Not Terms	88				119			
Avg. sem. score	1.563				2.887			
Avg. cos. dist.	0.734				0.753			
Pearson corr.	-0.010				0.341			
Spearman corr.	-0.110				0.208			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	18	20	21	29	30	25	32	32

were manually labeled as term-term pairs by at least one evaluator which suggests that, at least for English, embeddings can be used for extending the term list. Out of these 112 related terms, 52 were labeled as karst specific terms by at least one evaluator. For Slovenian, the results are worse, since out of 188 term-word pairs only 69 were labeled as term-term pairs and out of these only 36 are karst specific.

Out of 112 English term-term pairs, 62 were ranked first and second and 50 were ranked tenth and twentieth according to the cosine distance. Out of 69 Slovenian term-term pairs, 39 were ranked first or second and 30 were ranked as tenth or twentieth. This suggests that words that have most similar embeddings to terms according to the cosine distance (rank 1 and rank 2) are also more likely to be terms themselves than words that have less similar embeddings (rank 10 and rank 20). Similar applies to karst specific term-term pairs, where for English 30 were ranked first or second and 22 were ranked tenth or twentieth. For Slovenian, 24 out of 36 were ranked first or second and 12 were ranked tenth or twentieth.

When it comes to semantic similarity, unsurprisingly better ranked related words were manually evalu-

ated as semantically more similar. For example, the first ranked (most similar to terms according to the cosine distance) English related words got an average semantic similarity score[15] of 4.040 out of ten and first ranked Slovenian related words got an average semantic similarity score of 4.468. These are larger averages than semantic similarity score averages of 2.610 and 3.064 for English and Slovenian related words ranked as twentieth. Another interesting observation is the fact that the average semantic similarity score is the largest for English karst specific term-term pairs (5.702) and much lower if all the term-word pairs are considered (3.325). If we consider all term-term pairs, the average semantic similarity score is 4.710. Same applies for Slovenian term-word pairs, with semantic similarity score average rising from 3.859, when all term-words pairs are considered, to 5.536, when only term-term pairs are considered, and up to 6.722, when only karst specific term-term pairs are considered.

We also measure correlation between cosine distances and the semantic similarity scores for term-word pairs using Pearson and Spearman correlation coeficients. The correlation is generally low, the highest correlation being measured for Slovenian Karst specific term-term pairs where the Pearson correlation reached the value of 0.341 and Spearman the value of 0.208. There was no correlation measured on Slovene term-term pairs and surprisingly, a small negative Pearson correlation was measured on Slovenian karst specific term-term pairs and a small negative Spearman correlation was measured on English pairs which were labeled as terms. For more details see Appendix D.

5 Associated outputs

The work described in this deliverable has resulted in the following resources:

Description	URL	Availability
Code for RaKUn	`https://github.com/EMBEDDIA/RaKUn`	Public (GPL3)
Code for Term Alignment	`https://github.com/EMBEDDIA/4real2018`	Public (GPL3)
Code for TNT-KID	`https://github.com/EMBEDDIA/TNT_KID`	To become public*

*Will become public (licence TBD) after research publication.

Parts of this work are also described in detail in the following publications, which are attached to this deliverable as appendices:

Citation	Status	Appendix
Škrlj, B., Repar, A., Pollak, S. (2019). RaKUn: Rank-based keyword extraction via unsupervised learning and meta vertex aggregation. In Proceedings of the international conference on statistical language and speech processing SLSP 2019 (pp. 311–323). Springer.	Published	Appendix A
Repar, A., Martinc, M., Pollak, S. (2019, Nov). Reproduction, replication, analysis and adaptation of a term alignment approach. Language Resources and Evaluation. doi: 10.1007/s10579-019-09477-1.	Published	Appendix B
Repar, A., Podpečan, V., Vavpetič, A., Lavrač, N., Pollak, S. (2019). TermEnsembler: An ensemble learning approach to bilingual term extraction and alignment. Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication 25 (1), 93–120.	Published	Appendix C
Pollak, S., Repar, A., Martinc, M., Podpečan, V. (2019). Karst exploration: Extracting terms and definitions from Karst domain corpus. In Electronic lexicography in the 21st century: Proceedings of eLex 2019 conference.	Published	Appendix D

[15]Semantic similarity score for each related word is calculated as an average between the two semantic similarity scores given by two evaluators.

6 Conclusions and further work

In this report we presented the work performed during the first year in the scope of the Task 2.2. The main contributions are a novel unsupervised graph-based keyword extraction technique RaKUn, and especially TNT-KID, a completely novel neural tag detector developed in-house that we plan to publish in near future. We have described our attempts at reproducing the YAKE's results, and reported on an extensive empirical evaluation of YAKE along with RaKUn and other keyword detectors that for the first time offer insight into the difference between neural and non-neural approaches at such scale. In addition, several contributions to the field of terminology extraction and alignment have been proposed, such as a reimplementation study of a term alignment approach that can be used for keyword matching in a multilingual setting, as well as term expansion techniques. We have already started testing the approaches on the EMBEDDIA datasets, but will extend the work by testing TNT-KID on the Styria dataset, as well as an Estonian dataset. For Finnish partner STT, the work will be adapted in future, as they use the standardized IPTC keyword-tagging[16]. Being neural network based, especially TNT-KID offers a plethora of options for extension to a cross-lingual setting, and we plan to investigate these in future work.

[16]IPTC (International Press Telecommunications Council) develops and promotes technical standards to improve the management and exchange of information between content providers, intermediaries and consumers. Its members include news agencies, publishers and industry vendors. STT uses the system and every news article should have at least one (but preferably more) of the IPTC-keywords included.

Bibliography

Aker, A., Paramita, M., & Gaizauskas, R. (2013). Extracting bilingual terminologies from comparable corpora. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)* (Vol. 1, pp. 402–411).

Amjadian, E., Inkpen, D., Paribakht, T. S., & Faez, F. (2016). Local-global vectors to improve unigram terminology extraction. In *Proceedings of the 5th international workshop on computational terminology* (p. 2).

Arčan, M., Turchi, M., Tonelli, S., & Buitelaar, P. (2014, 10). Enhancing statistical machine translation with bilingual terminology in a CAT environment.. doi: https://doi.org/10.13140/2.1.1019.8404

Bader, B. W., & Chew, P. A. (2008). Enhancing multilingual latent semantic analysis with term alignment information. In *Proceedings of the 22nd international conference on computational linguistics-volume 1* (pp. 49–56).

Baevski, A., & Auli, M. (2018). Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*.

Bahmani, B., Chowdhury, A., & Goel, A. (2010). Fast incremental and personalized pagerank. *Proceedings of the VLDB Endowment*, *4*(3), 173–184.

Baisa, V., Ulipová, B., & Cukr, M. (2015). Bilingual terminology extraction in Sketch Engine. In *9th workshop on recent advances in slavonic natural language processing* (pp. 61–67).

Baldwin, T., & Tanaka, T. (2004). Translation by machine of complex nominals: Getting it right. In *Proceedings of the workshop on multiword expressions: Integrating processing* (pp. 24–31).

Beliga, S., Meštrović, A., & Martinčić-Ipšić, S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of information and organizational sciences*, *39*(1), 1–20.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bouamor, D., Semmar, N., & Zweigenbaum, P. (2013). Context vector disambiguation for bilingual lexicon extraction from comparable corpora. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 759–764).

Bougouin, A., Boudin, F., & Daille, B. (2013). TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the international joint conference on natural language processing (ijcnlp)* (pp. 543–551).

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, *509*, 257–289.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., & Jatowt, A. (2018a). A text feature based automatic keyword extraction method for single documents. In G. Pasi, B. Piwowarski, L. Azzopardi, & A. Hanbury (Eds.), *Advances in information retrieval* (pp. 684–691). Cham: Springer International Publishing.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., & Jatowt, A. (2018b). Yake! collection-independent automatic keyword extractor. In *European conference on information retrieval* (pp. 806–810).

Cao, Y., & Li, H. (2002). Base noun phrase translation using web data and the EM algorithm. In *Proceedings of the 19th international conference on computational linguistics - volume 1* (pp. 1–7).

Chiao, Y.-C., & Zweigenbaum, P. (2002). Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of the 19th international conference on computational linguistics - volume 2* (pp. 1–5).

Dai, Z., Yang, Z., Yang, Y., Cohen, W. W., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Daille, B., Gaussier, E., & Langé, J.-M. (1994). Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th conference on computational linguistics - volume 1* (pp. 515–521).

Daille, B., & Morin, E. (2005). French-English terminology extraction from comparable corpora. In *Natural language processing – ijcnlp 2005* (pp. 707–718).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diaz, F., Mitra, B., & Craswell, N. (2016). Query expansion with locally-trained word embeddings. In *Proceedings of the 54th annual meeting of the association for computational linguistics* (p. 367–377). Berlin, Germany.

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology*, *9*(1), 99–115.

El-Beltagy, S. R., & Rafea, A. (2009). Kp-miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, *34*(1), 132–144.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, *20*(1), 116–131.

Foo, J. (2012). *Computational terminology: Exploring bilingual and monolingual term extraction* (Unpublished doctoral dissertation). Linköping University Electronic Press.

Fung, P., & Yee, L. Y. (1998). An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th international conference on computational linguistics - volume 1* (pp. 414–420).

Gale, W., & Church, K. (1993). A program for aligning sentences in bilingual corpora. *Computational linguistics*, *19*(1), 75–102.

Gao, Y., & Yuan, Y. (2019). Feature-less end-to-end nested term extraction. In *Ccf international conference on natural language processing and chinese computing* (pp. 607–616).

Gaussier, E. (1998). Flow network models for word alignment and terminology extraction from bilingual corpora. In *Proceedings of the 17th international conference on computational linguistics - volume 1* (pp. 444–450).

Gollapalli, S. D., Li, X.-L., & Yang, P. (2017). Incorporating expert knowledge into keyphrase extraction. In *Thirty-first aaai conference on artificial intelligence.*

Grave, E., Joulin, A., Cissé, M., Jégou, H., et al. (2017). Efficient softmax approximation for gpus. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1302–1310).

Gu, J., Lu, Z., Li, H., & Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Ha, L. A., Fern, G., Mitkov, R., & Corpas, G. (2008). Mutual bilingual terminology extraction. In *Proceedings of the sixth international conference on language resources and evaluation (lrec'08)* (pp. 1818–1824).

Haque, R., Penkale, S., & Way, A. (2014). Bilingual termbank creation via log-likelihood comparison and phrase-based statistical machine translation. In *Proceedings of the 4th international workshop on computational terminology (computerm)* (pp. 42–51).

Hazem, A., & Morin, E. (2016). Efficient data selection for bilingual terminology extraction from comparable corpora. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 3401–3411).

Hazem, A., & Morin, E. (2017). Bilingual word embeddings for bilingual terminology extraction from specialized comparable corpora. In *Proceedings of the 8th international joint conference on natural language processing (volume 1: Long papers)* (pp. 685–693).

Henry, S., Cuffy, C., & McInnes, B. T. (2018). Vector representations of multi-word terms for semantic relatedness. *Journal of biomedical informatics*, *77*, 111–119.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., ... Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 782–792).

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on empirical methods in natural language processing* (pp. 216–223).

Ideue, M., Yamamoto, K., Utiyama, M., & Sumita, E. (2011). A comparison of unsupervised bilingual term extraction methods using phrase tables. In *Proceedings of the 13th machine translation summit* (pp. 346–351).

Johnson, I., & Macphail, A. (2000). IATE–Inter-Agency Terminology Exchange: Development of a single central terminology database for the institutions and agencies of the European Union. In *Proceedings of the workshop on terminology resources and computation, lrec 2000 conference.*

Kageura, K., & Umino, B. (1996). Methods of automatic term recognition: A review. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *3*(2), 259–289.

Khurshid, A., Gillman, L., & Tostevin, L. (2000). Weirdness indexing for logical document extrapolation and retrieval. In *Proceedings of the eighth text retrieval conference (trec-8).*

Kim, S. N., Medelyan, O., Kan, M.-Y., & Baldwin, T. (2010). Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th international workshop on semantic evaluation* (pp. 21–26). Stroudsburg, PA, USA: Association for Computational Linguistics.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th machine translation summit* (Vol. 5, pp. 79–86).

Kontonatsios, G., Korkontzelos, I., Tsujii, J., & Ananiadou, S. (2014). Combining string and context similarity for bilingual term alignment from comparable corpora. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1701–1712).

Krapivin, M., Autaeu, A., & Marchese, M. (2009). *Large dataset for keyphrases extraction* (Tech. Rep.). University of Trento.

Kupiec, J. (1993). An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st annual meeting on association for computational linguistics* (pp. 17–22).

Lee, L., Aw, A., Zhang, M., & Li, H. (2010). Em-based hybrid model for bilingual terminology extraction from comparable corpora. In *Proceedings of the 23rd international conference on computational linguistics: Posters* (pp. 639–646).

Luan, Y., Ostendorf, M., & Hajishirzi, H. (2017). Scientific information extraction with semi-supervised neural tagging. *arXiv preprint arXiv:1708.06075*.

Macken, L., Lefever, E., & Hoste, V. (2013). Texsis: Bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *19*(1), 1–30.

Medelyan, O., Frank, E., & Witten, I. H. (2009). Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 3-volume 3* (pp. 1318–1327).

Meng, R., Yuan, X., Wang, T., Brusilovsky, P., Trischler, A., & He, D. (2019). Does order matter? an empirical study on generating multiple keyphrases as a sequence. *arXiv preprint arXiv:1909.03590*.

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., & Chi, Y. (2017). Deep keyphrase generation. *arXiv preprint arXiv:1704.06879*.

Meyers, A. L., He, Y., Glass, Z., Ortega, J., Liao, S., Grieve-Smith, A., ... Babko-Malaya, O. (2018). The termolator: terminology recognition based on chunking, statistical and search-based scores. *Frontiers in Research Metrics and Analytics*, *3*, 19.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404–411).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Morin, E., Daille, B., Takeuchi, K., & Kageura, K. (2007). Bilingual terminology mining-using brain, not brawn comparable corpora. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 664–671).

Morin, E., Daille, B., Takeuchi, K., & Kageura, K. (2008, October). Brains, not brawn: The use of smart comparable corpora in bilingual terminology mining. *ACM Trans. Speech Lang. Process.*, *7*(1), 1:1–1:23.

Nassirudin, M., & Purwarianti, A. (2015). Indonesian-Japanese term extraction from bilingual corpora using machine learning. In *Advanced computer science and information systems (icacsis), 2015 international conference on* (pp. 111–116).

Nguyen, T. D., & Kan, M.-Y. (2007). Keyphrase extraction in scientific publications. In *International conference on asian digital libraries* (pp. 317–326).

Pollak, S., Repar, A., Martinc, M., & Podpečan, V. (2019). Karst exploration: Extracting terms and definitions from Karst domain corpus. In *Electronic lexicography in the 21st century: proceedings of eLex 2019 conference.*

Pollak, S., Vavpetič, A., Kranjc, J., Lavrač, N., & Vintar, v. (2012). NLP workflow for on-line definition extraction from English and Slovene text corpora. In *11th conference on natural language processing, KONVENS 2012 - empirical methods in natural language processing, vienna, austria* (pp. 53–60).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8).

Rapp, R. (1999). Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th annual meeting of the association for computational linguistics on computational linguistics* (pp. 519–526).

Repar, A., Martinc, M., & Pollak, S. (2019, Nov). Reproduction, replication, analysis and adaptation of a term alignment approach. *Language Resources and Evaluation*. Retrieved from `https://doi.org/10.1007/s10579-019-09477-1` doi: 10.1007/s10579-019-09477-1

Repar, A., Podpečan, V., Vavpetič, A., Lavrač, N., & Pollak, S. (2019). TermEnsembler: An ensemble learning approach to bilingual term extraction and alignment. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *25*(1), 93–120.

Rigouts Terryn, A., Hoste, V., & Lefever, E. (2019, Mar 26). In no uncertain terms: a dataset for monolingual and multilingual automatic term extraction from comparable corpora. *Language Resources and Evaluation*. Retrieved from `https://doi.org/10.1007/s10579-019-09453-9` doi: 10.1007/s10579-019-09453-9

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1–20.

Škrlj, B., Repar, A., & Pollak, S. (2019). RaKUn: Rank-based keyword extraction via unsupervised learning and meta vertex aggregation. In *International conference on statistical language and speech processing* (pp. 311–323).

Spitz, A., & Gertz, M. (2018). Entity-centric topic extraction and exploration: A network-based approach. In *European conference on information retrieval* (pp. 3–15).

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., & Schlüter, P. (2013). DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the 8th international conference on language resources and evaluation (lrec'2012).*

Steinberger, R., Pouliquen, B., & Hagman, J. (2002). Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. *Computational Linguistics and Intelligent Text Processing*, 101–121.

Sterckx, L., Demeester, T., Deleu, J., & Develder, C. (2015). Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th international conference on world wide web* (pp. 121–122).

Tiedemann, J. (2012, may). Parallel data, tools and interfaces in opus. In N. C. C. Chair) et al. (Eds.), *Proceedings of the eight international conference on language resources and evaluation (lrec'12).* Istanbul, Turkey: European Language Resources Association (ELRA).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vintar, Š. (2010). Bilingual term recognition revisited: The bag-of-equivalents term alignment approach and its evaluation. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, *16*(2), 141–158.

Wan, X., & Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the aaai conference* (Vol. 8, pp. 855–860).

Wang, R., Liu, W., & McDonald, C. (2016). Featureless domain-specific term extraction with minimal labelled data. In *Proceedings of the australasian language technology association workshop 2016* (pp. 103–112).

Wermter, J., & Hahn, U. (2005). Paradigmatic modifiability statistics for the extraction of complex multi-word terms. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 843–850).

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (2005). Kea: Practical automated keyphrase extraction. In *Design and usability of digital libraries: Case studies in the asia pacific* (pp. 129–152). IGI Global.

Yuan, X., Wang, T., Meng, R., Thaker, K., Brusilovsky, P., He, D., & Trischler, A. (2019). One size does not fit all: Generating and evaluating variable number of keyphrases. *arXiv preprint arXiv:1810.05241*.

Zesch, T., & Gurevych, I. (2009). Approximate matching for evaluating keyphrase extraction. In *Proceedings of the international conference ranlp-2009* (pp. 484–489).

RaKUn: Rank-based Keyword extraction via Unsupervised learning and Meta vertex aggregation

Blaž Škrlj[1,2], Andraž Repar[1,2], and Senja Pollak[2,3]

[1] Jožef Stefan International Postgraduate School
[2] Jožef Stefan Institute, Slovenia
[3] Usher Institute, Medical School, University of Edinburgh, Edinburgh
{blaz.skrlj,senja.pollak@ijs.si}, repar.andraz@gmail.com

The final authenticated publication, published in SLSP 2019, is available online at 10.1007/978-3-030-31372-2_26.

Abstract. Keyword extraction is used for summarizing the content of a document and supports efficient document retrieval, and is as such an indispensable part of modern text-based systems. We explore how load centrality, a graph-theoretic measure applied to graphs derived from a given text can be used to efficiently identify and rank keywords. Introducing meta vertices (aggregates of existing vertices) and systematic redundancy filters, the proposed method performs on par with state-of-the-art for the keyword extraction task on 14 diverse datasets. The proposed method is unsupervised, interpretable and can also be used for document visualization.

Keywords: keyword extraction · graph applications · vertex ranking· load centrality · information retrieval

1 Introduction and related work

Keywords are terms (i.e. expressions) that best describe the subject of a document [2]. A good keyword effectively summarizes the content of the document and allows it to be efficiently retrieved when needed. Traditionally, keyword assignment was a manual task, but with the emergence of large amounts of textual data, automatic keyword extraction methods have become indispensable. Despite a considerable effort from the research community, state-of-the-art keyword extraction algorithms leave much to be desired and their performance is still lower than on many other core NLP tasks [13]. The first keyword extraction methods mostly followed a supervised approach [14,24,31]: they first extract keyword features and then train a classifier on a gold standard dataset. For example, KEA [31], a state of the art supervised keyword extraction algorithm is based

on the Naive Bayes machine learning algorithm. While these methods offer quite good performance, they rely on an annotated gold standard dataset and require a (relatively) long training process. In contrast, unsupervised approaches need no training and can be applied directly without relying on a gold standard document collection. They can be further divided into statistical and graph-based methods. The former, such as YAKE [7,6], KP-MINER [10] and RAKE [25], use statistical characteristics of the texts to capture keywords, while the latter, such as Topic Rank [3], TextRank [22], Topical PageRank [29] and Single Rank [30], build graphs to rank words based on their position in the graph. Among statistical approaches, the state-of-the-art keyword extraction algorithm is YAKE [7,6], which is also one of the best performing keyword extraction algorithms overall; it defines a set of five features capturing keyword characteristics which are heuristically combined to assign a single score to every keyword. On the other hand, among graph-based approaches, Topic Rank [3] can be considered state-of-the-art; candidate keywords are clustered into topics and used as vertices in the final graph, used for keyword extraction. Next, a graph-based ranking model is applied to assign a significance score to each topic and keywords are generated by selecting a candidate from each of the top-ranked topics. Network-based methodology has also been successfully applied to the task of topic extraction [28].

The method that we propose in this paper, RaKUn, is a graph-based keyword extraction method. We exploit some of the ideas from the area of graph aggregation-based learning, where, for example, graph convolutional neural networks and similar approaches were shown to yield high quality vertex representations by aggregating their neighborhoods' feature space [5]. This work implements some of the similar ideas (albeit not in a neural network setting), where redundant information is aggregated into meta vertices in a similar manner. Similar efforts were shown as useful for hierarchical subnetwork aggregation in sensor networks [8] and in biological use cases of simulation of large proteins [9].

The main contributions of this paper are as follows. The notion of load centrality was to our knowledge not yet sufficiently exploited for keyword extraction. We show that this fast measure offers competitive performance to other widely used centralities, such as for example the PageRank centrality (used in [22]). To our knowledge, this work is the first to introduce the notion of meta vertices with the aim of aggregating similar vertices, following similar ideas to the statistical method YAKE [7], which is considered a state-of-the-art for the keyword extraction. Next, as part of the proposed RaKUn algorithm we extend the extraction from unigrams also to bigram and threegram keywords based on load centrality scores computed for considered tokens. Last but not least, we demonstrate how arbitrary textual corpora can be transformed into weighted graphs whilst maintaining *global* sequential information, offering the opportunity to exploit potential context not naturally present in statistical methods.

The paper is structured as follows. We first present the text to graph transformation approach (Section 2), followed by the introduction of the RaKUn keyword extractor (Section 3). We continue with qualitative evaluation (Section 4)

and quantitative evaluation (Section 5), before concluding the paper in Section 6.

2 Transforming texts to graphs

We first discuss how the texts are transformed to graphs, on which RaKUn operates. Next, we formally state the problem of keyword extraction and discuss its relation to graph centrality metrics.

2.1 Representing text

In this work we consider *directed* graphs. Let $G = (V, E)$ represent a graph comprised of a set of vertices V and a set of edges ($E \subseteq V \times V$), which are ordered pairs. Further, each edge can have a real-valued weight assigned. Let $\mathcal{D}$ represent a document comprised of tokens $\{t_1, \ldots, t_n\}$. The order in which tokens in text appear is known, thus $\mathcal{D}$ is a totally ordered set. A potential way of constructing a graph from a document is by simply observing word co-occurrences. When two words co-occur, they are used as an edge. However, such approaches do not take into account the sequence nature of the words, meaning that the order is lost. We attempt to take this aspect into account as follows. The given corpus is traversed, and for each element t_i, its successor t_{i+1}, together with a given element, forms a directed edge $(t_i, t_{i+1}) \in E$. Finally, such edges are weighted according to the number of times they appear in a given corpus. Thus the graph, constructed after traversing a given corpus, consists of all local neighborhoods (order one), merged into a single joint structure. Global contextual information is potentially kept intact (via weights), even though it needs to be detected via network analysis as proposed next.

2.2 Improving graph quality by meta vertex construction

A naïve approach to constructing a graph, as discussed in the previous section, commonly yields noisy graphs, rendering learning tasks harder. Therefore, we next discuss the selected approaches we employ in order to reduce both the computational complexity and the spatial complexity of constructing the graph, as well as increasing its quality (for the given down-stream task).

First, we consider the following heuristics which reduce the complexity of the graph that we construct for keyword extraction: Considered token length (while traversing the document $\mathcal{D}$, only tokens of length $\mu > \mu_{\min}$ are considered), and next, lemmatization (tokens can be lemmatized, offering spatial benefits and avoiding redundant vertices in the final graph). The two modifications yield a potentially "simpler" graph, which is more suitable and faster for mining.

Even if the optional lemmatization step is applied, one can still aim at further reducing the graph complexity by merging similar vertices. This step is called *meta vertex construction*. The motivation can be explained by the fact, that even similar lemmas can be mapped to the same keyword (e.g., mechanic

and mechanical; normal and abnormal). This step also captures spelling errors (similar vertices that will not be handled by lemmatization), spelling differences (e.g., British vs. American English), non-standard writing (e.g., in Twitter data), mistakes in lemmatization or unavailable or omitted lemmatization step.

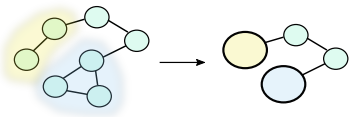

Fig. 1: Meta vertex construction. Sets of highlighted vertices are merged into a single vertex. The resulting graph has less vertices, as well as edges.

The meta-vertex construction step works as follows. Let V represent the set of vertices, as defined above. A meta vertex M is comprised of a set of vertices that are elements of V, i.e. $M \subseteq V$. Let M_i denote the i-th meta vertex. We construct a given M_i so that for each $u \in M_i$, u's initial edges (prior to merging it into a meta vertex) are rewired to the newly added M_i. Note that such edges connect to vertices which are not a part of M_i. Thus, both the number of vertices, as well as edges get reduced substantially. This feature is implemented via the following procedure:

1. Meta vertex candidate identification. Edit distance and word lengths distance are used to determine whether two words should be merged into a meta vertex (only if length distance threshold is met, the more expensive edit distance is computed).
2. The meta vertex creation. As common identifiers, we use the stemmed version of the original vertices and if there is more than one resulting stem, we select the vertex from the identified candidates that has the highest centrality value in the graph and its stemmed version is introduced as a novel vertex (meta vertex).
3. The edges of the words entailed in the meta vertex are next rewired to the meta vertex.
4. The two original words are removed from the graph.
5. The procedure is repeated for all candidate pairs.

A schematic representation of meta vertex construction is shown in Figure 1. The yellow and blue groups of vertices both form a meta vertex, the resulting (right) graph is thus substantially reduced, both with respect to the number of vertices, as well as the number of edges.

3 Keyword identification

Up to this point, we discussed how the graph used for keyword extraction is constructed. In this work, we exploit the notion of load centrality, a fast measure for estimating the importance of vertices in graphs. This metric can be defined as follows.

Load centrality The load centrality of a vertex falls under the family of centralities which are defined based on the number of shortest paths that pass through a given vertex v, i.e. $c(v) = \sum_{t \in V} \sum_{s \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}; t \neq s$, where $\sigma(s,t|v)$ represents the number of shortest paths that pass from vertex s to vertex t via v and $\sigma(s,t)$ the number of all shortest paths between s and t (see [4,11]). The considered load centrality measure is subtly different from the better known betweenness centrality; specifically, it is assumed that each vertex sends a package to each other vertex to which it is connected, with routing based on a priority system: given an input of flow x arriving at vertex v with destination v', v divides x equally among all neighbors of minimum shortest path to the target. The total flow passing through a given v via this process is defined as v's load. Load centrality thus maps from the set of vertices V to real values. For detailed description and computational complexity analysis, see [4]. Intuitively, vertices of the graph with the highest load centrality represent key vertices in a given network. In this work, we assume such vertices are good descriptors of the input document (i.e. keywords). Thus, ranking the vertices yields a priority list of (potential) keywords.

Formulating the RaKUn algorithm We next discuss how the considered centrality is used as part of the whole keyword extraction algorithm RaKUn, summarized in Algorithm 1. The algorithm consists of three main steps described next. First, a graph is constructed from a given ordered set of tokens (e.g., a document) (lines 1 to 8). The resulting graph is commonly very sparse, as most of the words rarely co-occur. The result of this step is a smaller, denser graph, where both the number of vertices, as well as edges is lower. Once constructed, load centrality (line 10) is computed for each vertex. Note that at this point, should the top k vertices by centrality be considered, only single term keywords emerge. As it can be seen from line 11, to extend the selection to 2- and 3-grams, the following procedure is proposed:

2-gram keywords. Keywords comprised of two terms are constructed as follows. First, pairs of first order keywords (all tokens) are counted. If the support (= number of occurrences) is higher than f (line 11 in Algorithm 1), the token pair is considered as potential 2-gram keyword. The load centralities of the two tokens are averaged, i.e. $c_v = \frac{c_1+c_2}{2}$, and the obtained keywords are considered for final selection along with the computed ranks.

3-gram keywords. For construction of 3-gram keywords, we follow a similar idea to that of bigrams. The obtained 2-gram keywords (previous step) are

Algorithm 1: RaKUn algorithm.

Data: Document D, consisting of n tokens $t_1, \ldots, t_n$
Parameters : General: number of keywords k, minimal token length μ; Meta vertex parameters: edit distance threshold α, word length difference threshold l, Multi-word keywords parameters: path length p, 2-gram frequency threshold f
Result: A set of keywords $\mathcal{K}$

corpusGraph← EmptyGraph; ▷ `Initialization.`
for $t_i \in D$ **do**
  edge ← (t_i, t_{i+1});
  if *edge not in corpusGraph and len*$(t_i) \geq \mu$ **then**
    add *edge* to *corpusGraph* ; ▷ `Graph construction.`
  end
  updateEdgeWeight(*corpusGraph*, *edge*) ; ▷ `Weight update.`
end
corpusGraph ← generateMetaVertices(*corpusGraph*, α, l);
tokenRanks ← loadCentrality(*corpusGraph*) ; ▷ `Initial token ranks.`
scoredKeywords ← generateKeywords(p, f, tokenRanks) ; ▷ `Keyword search.`
$\mathcal{K}$ = scoredKeywords[:k];
return $\mathcal{K}$

further explored as follows. For each candidate 2-gram keyword, we consider two extension scenarios: Extending the 2-gram from the left side. Here, the in-neighborhood of the left token is considered as a potential extension to a given keyword. Ranks of such candidates are computed by averaging the centrality scores in the same manner as done for the 2-gram case. Extending the 2-gram from the right side. The difference with the previous point is that all outgoing connections of the rightmost vertex are considered as potential extensions. The candidate keywords are ranked, as before, by averaging the load centralities, i.e. $c_v = \frac{1}{3}\sum_{i=1}^{3} c_i$.

Having obtained a set of (keyword, score) pairs, we finally sort the set according to the scores (descendingly), and take top k keywords as the result. We next discuss the evaluation the proposed algorithm.

4 Qualitative evaluation

RaKUn can be used also for visualization of keywords in a given document or document corpus. A visualization of extracted keywords is applied to an example from wiki20 [21] (for dataset description see Section 5.1), where we visualize both the global corpus graph, as well as a local (document) view where keywords are emphasized, see Figures 2 and 3, respectively. It can be observed that the global graph's topology is far from uniform — even though we did not perform any tests of scale-freeness, we believe the constructed graphs are subject to distinct topologies, where keywords play prominent roles.

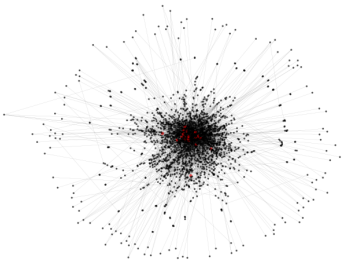

Fig. 2: Keyword visualization. Red dots represent keywords, other dots represent the remainder of the corpus graph.

5 Quantitative evaluation

This section discusses the experimental setting used to validate the proposed RaKUn approach against state-of-the-art baselines. We first describe the datasets, and continue with the presentation of the experimental setting and results.

5.1 Datasets

For RaKUn evaluation, we used 14 gold standard datasets from the list of [7,6], from which we selected datasets in English. Detailed dataset descriptions and statistics can be found in Table 1, while full statistics and files for download can be found online[4]. Most datasets are from the domain of computer science or contain multiple domains. They are very diverse in terms of the number of documents—ranging from *wiki20* with 20 documents to *Inspec* with 2,000 documents, in terms of the average number of gold standard keywords per document—from 5.07 in *kdd* to 48.92 in *500N-KPCrowd-v1.1*—and in terms of the average length of the documents—from 75.97 in *kdd* to *SemEval2017* with 8332.34.

5.2 Experimental setting

We adopted the same evaluation procedure as used for the series of results recently introduced by YAKE authors [6][5]. Five fold cross validation was used to determine the overall performance, for which we measured Precision, Recall and

[4] https://github.com/LIAAD/KeywordExtractor-Datasets

[5] We attempted to reproduce YAKE evaluation procedure based on their experimental setup description and also thank the authors for additional explanation regarding the evaluation. For comparison of results we refer to their online repository https://github.com/LIAAD/yake [7]

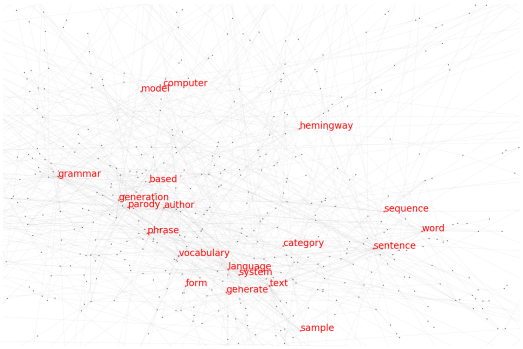

Fig. 3: Keyword visualization. A close-up view shows some examples of keywords and their location in the corpus graph. The keywords are mostly located in the central part of the graph.

F1 score, with the latter being reported in Table 2.[6] Keywords were stemmed prior to evaluation.[7] As the number of keywords in the gold standard document is not equal to the number of extracted keywords (in our experiments k=10), in the recall we divide the correctly extracted keywords by the number of keywords parameter k, if in the gold standard number of keywords is higher than k.

Selecting default configuration. First, we used a dedicated run for determining the default parameters. The cross validation was performed as follows. For each train-test dataset split, we kept the documents in the test fold intact, whilst performing a grid search on the train part to find the best parametrization. Finally, the selected configuration was used to extract keywords on the unseen test set. For each train-test split, we thus obtained the number of true and false positives, as well as true and false negatives, which were summed up and, after all folds were considered, used to obtain final F1 scores, which served for default parameter selection. The grid search was conducted over the following parameter range Num keywords: 10, Num tokens (the number of tokens a keyword can consist of): Count threshold (minimum support used to determine potential bigram candidates): Word length difference threshold (maximum difference in word length used to determine whether a given pair of words shall be aggregated): $[0, 2, 4]$, Edit length difference (maximum edit distance allowed to consider a given pair of words for aggregation): $[2, 3]$, Lemmatization: [yes, no].

Even if one can use the described grid-search fine-tunning procedure to select the best setting for individual datasets, we observed that in nearly all the cases the best settings were the same. We therefore selected it as the *default*, which

[6] The complete results and the code are available at `https://github.com/SkBlaz/rakun`

[7] This being a standard procedure, as suggested by the authors of YAKE.

Table 1: Selection of keyword extraction datasets in English language

Dataset	Desc.	No. docs	Avg. keywords	Avg. doc length
500N-KPCrowd-v1.1 [18]	Broadcast news transcriptions	500	48.92	408.33
Inspec [15]	Scientific journal papers from Computer Science collected between 1998 and 2002	2000	14.62	128.20
Nguyen2007 [23]	Scientific conference papers	209	11.33	5201.09
PubMed	Full-text papers collected from PubMed Central	500	15.24	3992.78
Schutz2008[26]	Full-text papers collected from PubMed Central	1231	44.69	3901.31
SemEval2010 [17]	Scientific papers from the ACM Digital Library	243	16.47	8332.34
SemEval2017 [1]	500 paragraphs selected from 500 ScienceDirect journal articles, evenly distributed among the domains of Computer Science, Material Sciences and Physics	500	18.19	178.22
citeulike180 [19]	Full-text papers from the CiteULike.org	180	18.42	4796.08
fao30 [20]	Agricultural documents from two datasets based on Food and Agriculture Organization (FAO) of the UN	30	33.23	4777.70
fao780 [20]	Agricultural documents from two datasets based on Food and Agriculture Organization (FAO) of the UN	779	8.97	4971.79
kdd [12]	Abstracts from the ACM Conference on Knowledge Discovery and Data Mining (KDD) during 2004-2014	755	5.07	75.97
theses100	Full master and Ph.D. theses from the University of Waikato	100	7.67	4728.86
wiki20 [21]	Computer science technical research reports	20	36.50	6177.65
www [12]	Abstracts of WWW conference papers from 2004-2014	1330	5.80	84.08

can be used also on new unlabeled data. The default parameter setting was as follows. The number of tokens was set to 1, Count threshold was thus not needed (only unigrams), for meta vertex construction Word length difference threshold was set to 3 and Edit distance to 2. Words were initially lemmatized. Next, we report the results using these selected parameters (same across all datasets), by which we also test the general usefulness of the approach.

5.3 Results

The results are presented in Table 2, where we report on F1 with the default parameter setting of RaKUn, together with the results from related work, as reported in the github table of the YAKE [7][8]. We first observe that on the selection of datasets, the proposed RaKUn outperforms (on average) any other graph-based method. We also see that it performs better on a subset of datasets (discussed next), yet not overalls. Such results demonstrate that the proposed method finds keywords differently, indicating load centrality, combined with meta vertices, represents a promising research venue. RaKUn performs well on *citeulike180* and similar single-keyword datasets, which is why the default configuration (which returns unigrams only) was able to perform well.

Similarly, four of the five well-performing datasets (*Schutz2008*, *fao30*, *citeulike180*, *wiki20*) include long documents (more than 3,900 words), with the exception being *500N-KPCrowd-v1.1*. For details, see Table 1. We observe that the proposed RaKUn outperforms the majority of other competitive graph-based methods. For example, the most similar variants Topical PageRank and TextRank do not perform as well on the majority of the considered datasets.

[8] https://github.com/LIAAD/yake/blob/master/docs/YAKEvsBaselines.jpg (accessed on: June 11, 2019)

Table 2: Performance comparison with state-of-the-art approaches.

Dataset	RaKUn	YAKE	Single Rank	KEA	KP-MINER	Text Rank	Topic Rank	Topical PageRank
500N-KPCrowd-v1.1	0.167	**0.173**	0.157	0.159	0.093	0.111	0.172	0.158
Inspec	0.076	0.316	**0.378**	0.150	0.047	0.098	0.289	0.361
Nguyen2007	0.096	0.256	0.158	0.221	**0.314**	0.167	0.173	0.148
PubMed	0.095	0.106	0.039	**0.216**	0.114	0.071	0.085	0.052
Schutz2008	0.221	0.196	0.086	0.182	0.230	0.118	0.258	0.123
SemEval2010	0.152	0.211	0.129	0.215	**0.261**	0.149	0.195	0.125
SemEval2017	0.162	0.329	**0.449**	0.201	0.071	0.125	0.332	0.443
citeulike180	0.240	0.256	0.066	**0.317**	0.240	0.112	0.156	0.072
fao30	0.165	**0.184**	0.066	0.139	0.183	0.077	0.154	0.107
fao780	0.112	**0.187**	0.085	0.114	0.174	0.083	0.137	0.108
kdd	0.092	**0.156**	0.085	0.063	0.036	0.050	0.055	0.089
theses100	0.069	0.111	0.060	0.104	**0.158**	0.058	0.114	0.083
wiki20	0.093	**0.162**	0.038	0.134	0.156	0.074	0.106	0.059
www	0.082	**0.172**	0.097	0.072	0.037	0.059	0.067	0.101
#Top 3	5	11	3	5	9	0	5	4

6 Conclusions and further work

In this work we proposed RaKUn, a novel unsupervised keyword extraction algorithm which exploits the efficient computation of load centrality, combined with the introduction of meta vertices, which notably reduce corpus graph sizes. The method is fast, and performs well compared to state-of-the-art such as YAKE and graph-based keyword extractors. In further work, we will test the method on other languages. We also believe additional semantic background knowledge information could be used to prune the graph's structure even further, and potentially introduce keywords that are inherently not even present in the text (cf.[27]). The proposed method does not attempt to exploit meso-scale graph structure, such as convex skeletons or communities, which are known to play prominent roles in real-world networks and could allow for vertex aggregation based on additional graph properties. We believe the proposed method could also be extended using the Ricci-Oliver [16] flows on weighted graphs.

Acknowledgements The work was supported by the Slovenian Research Agency through a young researcher grant [BŠ], core research programme (P2-0103), and projects Semantic Data Mining for Linked Open Data (N2-0078) and Terminology and knowledge frames across languages (J6-9372). This work was supported also by the EU Horizon 2020 research and innovation programme, Grant No. 825153, EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflect only the authors views and the EC is not responsible for any use that may be made of the information it contains. We also thank the authors of YAKE for their clarifications.

References

1. Augenstein, I., Das, M., Riedel, S., Vikraman, L., McCallum, A.: Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. CoRR **abs/1704.02853** (2017)
2. Beliga, S., Meštrović, A., Martinčić-Ipšić, S.: An overview of graph-based keyword extraction methods and approaches. Journal of information and organizational sciences **39**(1), 1–20 (2015)
3. Bougouin, A., Boudin, F., Daille, B.: Topicrank: Graph-based topic ranking for keyphrase extraction. In: International Joint Conference on Natural Language Processing (IJCNLP). pp. 543–551 (2013)
4. Brandes, U.: On variants of shortest-path betweenness centrality and their generic computation. Social Networks **30**(2), 136–145 (2008)
5. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering **30**(9), 1616–1637 (2018)
6. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A.M., Nunes, C., Jatowt, A.: A text feature based automatic keyword extraction method for single documents. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.) Advances in Information Retrieval. pp. 684–691. Springer International Publishing, Cham (2018)
7. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A.M., Nunes, C., Jatowt, A.: Yake! collection-independent automatic keyword extractor. In: European Conference on Information Retrieval. pp. 806–810. Springer (2018)
8. Chan, H., Perrig, A., Song, D.: Secure hierarchical in-network aggregation in sensor networks. In: Proceedings of the 13th ACM conference on Computer and communications security. pp. 278–287. ACM (2006)
9. Doruker, P., Jernigan, R.L., Bahar, I.: Dynamics of large proteins through hierarchical levels of coarse-grained structures. Journal of computational chemistry **23**(1), 119–127 (2002)
10. El-Beltagy, S.R., Rafea, A.: Kp-miner: A keyphrase extraction system for english and arabic documents. Information Systems **34**(1), 132–144 (2009)
11. Goh, K.I., Kahng, B., Kim, D.: Universal behavior of load distribution in scale-free networks. Phys. Rev. Lett. **87**, 278701 (Dec 2001)
12. Gollapalli, S.D., Caragea, C.: Extracting keyphrases from research papers using citation networks. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
13. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: A survey of the state of the art. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1262–1273 (2014)
14. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing. pp. 216–223. Association for Computational Linguistics (2003)
15. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. pp. 216–223. EMNLP '03 (2003)
16. Jin, M., Kim, J., Gu, X.D.: Discrete surface ricci flow: Theory and applications. In: IMA International Conference on Mathematics of Surfaces. pp. 209–232. Springer (2007)
17. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 21–26. SemEval '10 (2010)

18. Marujo, L., Viveiros, M., da Silva Neto, J.P.: Keyphrase cloud generation of broadcast news. CoRR **abs/1306.4606** (2013)
19. Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3. pp. 1318–1327. EMNLP '09 (2009)
20. Medelyan, O., Witten, I.H.: Domain-independent automatic keyphrase indexing with small training sets. Journal of the American Society for Information Science and Technology **59**(7), 1026–1040
21. Medelyan, O., Witten, I.H., Milne, D.: Topic indexing with wikipedia. In: Proceedings of the AAAI WikiAI workshop. vol. 1, pp. 19–24 (2008)
22. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing (2004)
23. Nguyen, T.D., Kan, M.Y.: Keyphrase extraction in scientific publications. In: Goh, D.H.L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers. pp. 317–326. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
24. Nguyen, T.D., Luong, M.T.: Wingnus: Keyphrase extraction utilizing document logical structure. In: Proceedings of the 5th international workshop on semantic evaluation. pp. 166–169. Association for Computational Linguistics (2010)
25. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. Text mining: applications and theory pp. 1–20 (2010)
26. Schutz, A.T., et al.: Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. Master's thesis, National University of Ireland (2008)
27. Škrlj, B., Kralj, J., Lavrač, N., Pollak, S.: Towards robust text classification with semantics-aware recurrent neural architecture. Machine Learning and Knowledge Extraction **1**(2), 575–589 (2019)
28. Spitz, A., Gertz, M.: Entity-centric topic extraction and exploration: A network-based approach. In: European Conference on Information Retrieval. pp. 3–15. Springer (2018)
29. Sterckx, L., Demeester, T., Deleu, J., Develder, C.: Topical word importance for fast keyphrase extraction. In: Proceedings of the 24th International Conference on World Wide Web. pp. 121–122. ACM (2015)
30. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: AAAI. vol. 8, pp. 855–860 (2008)
31. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: Practical automated keyphrase extraction. In: Design and Usability of Digital Libraries: Case Studies in the Asia Pacific, pp. 129–152. IGI Global (2005)

Lang Resources & Evaluation
https://doi.org/10.1007/s10579-019-09477-1

ORIGINAL PAPER - REPLICABILITY & REPRODUCIBILITY

Reproduction, replication, analysis and adaptation of a term alignment approach

Andraž Repar[1,2] · **Matej Martinc**[1,2] · **Senja Pollak**[2,3]

Abstract In this paper, we look at the issue of reproducibility and replicability in bilingual terminology alignment (BTA). We propose a set of best practices for reproducibility and replicability of NLP papers and analyze several influential BTA papers from this perspective. Next, we present our attempts at replication and reproduction, where we focus on a bilingual terminology alignment approach described by Aker et al. (Extracting bilingual terminologies from comparable corpora. In: Proceedings of the 51st annual meeting of the association for computational linguistics, vol. 1 402–411, 2013) who treat bilingual term alignment as a binary classification problem and train an SVM classifier on various dictionary and cognate-based features. Despite closely following the original paper with only minor deviations—in areas where the original description is not clear enough—we obtained significantly worse results than the authors of the original paper. We then analyze the reasons for the discrepancy and describe our attempts at adaptation of the approach to improve the results. Only after several adaptations, we achieve results which are close to the results published in the original paper. Finally, we perform the experiments to verify the replicability and reproducibility of our own code. We publish our code and datasets online to assure the reproducibility of the results of our experiments and implement the selected BTA models in an online

✉ Andraž Repar
repar.andraz@gmail.com

Matej Martinc
matej.martinc@ijs.si

Senja Pollak
senja.pollak@ijs.si

[1] Jožef Stefan Postgraduate School, Ljubljana, Slovenia

[2] Jožef Stefan Institute, Ljubljana, Slovenia

[3] Usher institute, Medical school, University of Edinburgh, Edinburgh, UK

Published online: 18 November 2019

platform making them easily reusable even by the technically less-skilled researchers.

Keywords Bilingual term alignment · Reproducibility · Machine learning · Cognates

1 Introduction

The issue of reproducibility has been on the radar of researchers at least for the past 25 years, particularly in the life science research (e.g. Yentis et al. 1993; Prinz et al. 2011; Camerer et al. 2016). More recently, many other disciplines have started to acknowledge the crisis of reproducibility, among them also human language technology research (Pedersen 2008; Kano et al. 2009; Fokkens et al. 2013; Branco et al. 2017; Wieling et al. 2018). However, the basic terminology has remained confusing with different authors using different terms for the same concepts which is why Cohen et al. (2018) describe the three dimensions of reproducibility in natural language processing (NLP) and provide a set of definitions for the various concepts used when discussing reproducibility in NLP. They first differentiate between the concepts of **replicability** (or repeatability), which they define as *the ability to repeat the experiment described in a study*, and **reproducibility**, which describes the outcome—whether *the replicability efforts lead to the same conclusions.* Then they further break down reproducibility into reproducibility of a **conclusion** (defined as an explicit statement in the paper arrived at on the basis of the results of the experiments), reproducibility of a **finding** (a relationship between the values for some reported figure of merit) and reproducibility of a **value** (actual measured or calculated numbers).

In this paper we extend our reproducibility study (Repar et al. 2018), presented at the Workshop on Research Results Reproducibility and Resources Citation (4REAL Workshop, Branco et al. (2018)) organized within the scope of the 11th Language Resources and Evaluation Conference (LREC 2018). Our original motivation came from our interest and need for a terminology alignment tool, and the paper by Aker et al. (2013) titled "Extracting Bilingual Terminology from Parallel Corpora" seemed a perfect candidate for reproduction with nearly perfect results, coverage of the Slovenian-English pair (which were the languages of our interest) and what seemed like a well described and simple to replicate method. The authors treat aligning terms in two languages as a binary classification problem. They use an SVM binary classifier (Joachims 2002) and training data terms taken from the Eurovoc thesaurus (Steinberger et al. 2002) and construct two types of features: dictionary-based (using word alignment dictionaries created with Giza++ (Och and Ney 2003)) and cognate-based (effectively utilizing the similarity of terms across languages). Given that the results looked very promising—precision on the held-out set was 1 or close to 1 for many language pairs, we thought we could use the approach in our work and we set out to replicate it. We expected a straightforward process, but it turned out to be anything but: the results of our experiments were very vastly different from the original paper. For example, while the original paper

reports an extremely high precision (1 or close to 1) for the language pairs we have focused on, our experiments showed a precision below 0.05. Based on the reproducibility dimensions mentioned above, in our original reproducibility experiment from Repar et al. (2018) we were not able to reproduce any of the three dimensions: the values and findings in our experiments were vastly different, and—had we stopped at this point—we would have concluded that the proposed machine learning approach is not suitable for bilingual terminology alignment. Only after a great deal of tweaking and optimization have we managed to get to a respectable precision level (similar to the results in the original paper).

In the present paper, we aim to explore the issue of reproducibility and replicability in the field of terminology alignment further. To do so, we extend the work in Repar et al. (2018) with the following:

- an overview of bilingual terminology extraction and alignment approaches in terms of replicability and reproducibility.
- extending the original reproducibility experiment to two additional languages, resulting in Slovenian, French and Dutch as target languages from three different language families.
- providing very detailed description of feature construction.
- additional filtering and refinement of the cognate-based features.
- a reproducibility experiment with source code from Repar et al. (2018).
- implementation of our code into an online data mining platform ClowdFlows.
- a discussion on good practices for reproducibility and replicability in NLP.

This paper is organized as follows: After the introduction in Sect. 1, we present the related work and the analysis of bilingual terminology alignment papers from the point of view of replicability and reproducibility (Sect. 2). Section 3 contains the main replicability and reproducibility experiments, and is followed by Sect. 4, which describes our attempts at improving the results of the replicated approach, while Sect. 5 contains the results of manual evaluation. Section 6 describes the reproducibility experiment using our code from Repar et al. (2018) and Sect. 7 the implementation of the system in the ClowdFlows platform, for making it accessible to a wider community. Section 8 contains the conclusions and presents ideas for future work. The code and datasets of our experiments are published online, to enable future reproducibility and replicability.[1]

2 Overview of bilingual terminology extraction and alignment approaches

In this section we first look at the related work on bilingual terminology extraction and alignment and then analyze several related papers from the viewpoint of replicability and reproducibility.

[1] http://source.ijs.si/mmartinc/4real2018.

2.1 Related work

We start by providing a clarification regarding the terminology used in this paper. Following the distinction between two basic approaches made by Foo (2012):

- *extract-align* where we first extract monolingual candidate terms from both sides of the corpus and then align the terms, and
- *align-extract* where we first align single and multi-word units in parallel sentences and then extract the relevant terminology from a list of candidate term pairs.

we propose the following two definitions:

- *Bilingual terminology extraction* is the process which, given the input of related specialized monolingual corpora, results in the output of terms aligned between two languages. The process can either start with extracting monolingual candidate terms and aligning them between two languages (i.e. extract-align) or with aligning phrases and then extracting terms (i.e. align-extract) or any other sequence of actions.
- *Bilingual terminology alignment* is the process of aligning terms between two candidate term lists in two languages.

Bilingual terminology alignment has a narrower focus than bilingual terminology extraction, but the two terms are often used interchangeably in various papers. For example, the title of the paper we were trying to replicate "Extracting bilingual terminologies from comparable corpora" is somewhat misleading in this regard, since the paper primarily deals with bilingual terminology alignment, while they utilize monolingual terminology extraction (specifically the approach by Pinnis et al. (2012) without any modifications) only in the manual evaluation experiments.

The primary purpose of bilingual terminology extraction is to build a term bank—i.e. a list of terms in one language along with their equivalents in the other language. With regard to the input text, we can distinguish between alignment on the basis of a parallel corpus and alignment on the basis of a comparable corpus. For the translation industry, bilingual terminology extraction from parallel corpora is extremely relevant due to the large amounts of sentence-aligned parallel corpora available in the form of translation memories (in the TMX file format). Consequently, initial attempts at bilingual terminology extraction involved parallel input data (Kupiec 1993; Daille et al. 1994; Gaussier 1998), and the interest of the community continued until today (Ha et al. 2008; Ideue et al. 2011; Macken et al. 2013; Haque et al. 2014; Arčan et al. 2014; Baisa et al. 2015). However, most parallel corpora are owned by private companies,[2] such as language service providers, who consider them to be their intellectual property and are reluctant to share them publicly. For this reason (and in particular for language pairs not

[2] However, some publicly available parallel corpora do exist. A good overview can be found at the OPUS web portal (Tiedemann 2012).

involving English) considerable efforts have also been invested into researching bilingual terminology extraction from comparable corpora (Fung and Yee 1998; Rapp 1999; Chiao and Zweigenbaum 2002; Cao and Li 2002; Daille and Morin 2005; Morin et al. 2008; Vintar 2010; Bouamor et al. 2013; Hazem and Morin 2016, 2017).

Despite the problem of bilingual term alignment lending itself well to the binary classification task, there have been relatively few approaches utilizing machine learning. For example, similar to Aker et al. (2013), Baldwin and Tanaka (2004) generate corpus-based, dictionary-based and translation-based features and train an SVM classifier to rank the translation candidates. Note that they only focus on multi-word noun phrases (noun + noun). A similar approach, again focusing on noun phrases, is also described by Cao and Li (2002). Finally, Nassirudin and Purwarianti (2015) also reimplement Aker et al. (2013) for the Indonesian-Japanese language pair and further expand it with additional statistical features. In the best scenario, their accuracy, precision and recall all exceed 90% but the results are not directly comparable since Nassirudin and Purwarianti (2015) use tenfold cross-validation while Aker et al. (2013) use a held-out test set. In addition, Nassirudin and Purwarianti (2015) have a balanced test set while Aker et al. (2013) use a very unbalanced one (ratio of positive vs. negative examples 1:2000).

2.2 Analysis of past papers on bilingual terminology extraction from the viewpoint of reproducibility and replicability

In an ideal reproducibility and replicability scenario, a scientific paper would contain an accurate and clear description of the datasets used and experiments conducted and the authors would provide a single link containing all the datasets (versions, subsets etc.) used for the experiments along with the experiment source code (or alternatively, an online tool to run the experiments). These could then be used to replicate the experiments and reproduce the results using the descriptions provided in the paper.

We have analyzed several[3] bilingual terminology extraction papers from the past 25 years from the point of view of dataset, code and tool availability. The summary of results is available in Table 1.

2.2.1 Dataset availability

In terms of dataset availability, we looked at whether the paper contains some description of how the datasets were constructed and which could (theoretically) be used to reconstruct the datasets. Note that under "dataset", we include corpora, gold

[3] The selection process was as follows: the starting point were selected seminal papers on the field, as well as two queries in the ACL Anthology database: "term alignment" and "bilingual terminology extraction". We analyzed the papers found by these two queries as well as additional papers mentioned in the related works sections of these papers and the main criterion for including a paper in our analysis was that it primarily deals with bilingual terminology extraction (and not for example latent semantic analysis, such as Bader and Chew (2008)). However, no strict systematic review with inclusion and exclusion criteria was made, as such a survey would be beyond the needs of this paper.

Table 1 An analysis of bilingual terminology extraction papers from the point of view of reproducibility and replicability

Paper	Dataset	Code	Tool	Google Scholar citations as of September 2019
Kupiec (1993)	Links	No	No	333
Daille et al. (1994)	No	No	No	268
Fung and Yee (1998)	Description	No	No	427
Gaussier (1998)	No	No	No	84
Rapp (1999)	Description	No	No	552
Chiao and Zweigenbaum (2002)	Description	No	No	135
Cao and Li (2002)	Description	No	No	141
Morin et al. (2007)	No	No	No	113
Daille and Morin (2005)	Obsolete	No	Obsolete	56
Morin et al. (2008)	Links	No	Obsolete	22
Ha et al. (2008)	Description	No	No	4
Lee et al. (2010)	Description	No	No	22
Vintar (2010)	No	No	Obsolete	53
Ideue et al. (2011)	No	No	Yes[a]	9
Macken et al. (2013)	No	No	No	48
Bouamor et al. (2013)	Description	No	No	24
Aker et al. (2013)	Links	No	No	36
Arčan et al. (2014)	Links	No	No	18
Haque et al. (2014)	Links	No	No	11
Kontonatsios et al. (2014)	Description	No	No	14
Baisa et al. (2015)	No	No	Yes	5
Hazem and Morin (2016)	Links	No	No	12
Hazem and Morin (2017)	Links	No	No	2

[a]A Perl module (Term Extract) was used, however the link leads to a Japanese website

standard termlists, seed dictionaries and all other linguistic resources needed to conduct the experiments in the paper. For example, we consider the following paragraph from Rapp (1999) to be a valid description of a dataset: *As the German corpus, we used 135 million words of the newspaper Frankfurter Allgemeine Zeitung (1993 to 1996), and as the English corpus 163 million words of the Guardian (1990 to 1994).* On the other hand, this paragraph from Ideue et al. (2011) is not considered a valid description: *We extracted bilingual term candidates from a Japanese-English parallel corpus consisting of documents related to apparel products.* In the former example, dataset reconstruction would be difficult but not impossible, while in the latter it is impossible. An even better option is to link to actual datasets or refer to papers where datasets are described and linked, which is why we also looked for dataset links and/or references in the analyzed papers. Note that there are several examples where links are provided only for a selection of the datasets used in the experiments (e.g., Morin et al. (2008)).

As evident from Table 1, dataset availability is the least problematic aspect of reproducibility and replicability in terminology (extraction and) alignment papers with approximately two thirds of the analyzed papers (15 out of 23) either containing a description of the resources used for the experiments, providing links to them or refering to papers where they are described.

We expected the earlier papers to have less information on datasets than latter ones, but this turned out not to be the case. In fact, the earliest paper analyzed—Kupiec (1993)—provides a reference to a publicly available corpus (Canadian Hansards (Gale and Church 1993)). The first paper to have a separate section with data/resource description is Rapp (1999) and from this point on, almost all papers have such a section—usually titled "Data and Resources", "Resources and Experimental Setup", "Linguistic resources" or similar.

However, it is rarely documented what version of the dataset was used and whether an entire dataset was used or only a part of it (as in random selection, train-test split, etc.). In most cases, little information is provided on the actual subsets used for the experiments. Another aspect of dataset use is the languages: when one of the languages involved is English, it is much easier to find datasets than for other language combinations. Finally, there is also the issue of keeping the links active. For example, many of the links in Daille and Morin (2005) and Morin et al. (2008) are not active anymore while Bouamor et al. (2013) state that the corpora and terminology gold standard lists created for the paper will be shared publicly, but no links are provided.

The most significant problem encountered during our analysis was the fact that terminology alignment is most often not the sole focus of a paper, such as in Haque et al. (2014), where the experiments start with monolingual terminology extraction from two languages and the extracted terms are then aligned. As terminology extraction and alignment go hand-in-hand, it may often be impossible to make a clear distinction between the terminology extraction and terminology alignment datasets. This means that the dataset results in Table 1 are not a true apple-to-apple comparison: one paper might link to the parallel corpus used to extract terms from, while another to a gold standard termlist. Our main criterion was whether the dataset description (or link) could be used to replicate the experiments described in the paper.

An ideal terminology (extraction and) alignment dataset would therefore consist of a bilingual or multilingual (parallel or comparable) corpus along with reference (gold standard) term lists containing terms that can be found in the corpus. Such corpora are TTC wind energy and TC mobile technology[4], which contain data for six languages (English, French, German, Spanish, Russian, Latvian, Chinese), or the Bitter corpus[5], which contains data for the EN-IT language pair. The first was used in Hazem and Morin (2016), while the second one by Arčan et al. (2014). Since such datasets are scarce, researchers employ various methodologies for constructing their own datasets. One method, used by Aker et al. (2013), is to take one of the available multilingual translation memories containing EU documentation (such as

[4] http://www.lina.univ-nantes.fr/?Reference-Term-Lists-of-TTC.html.

[5] https://hlt-mt.fbk.eu/technologies/bittercorpus.

Europarl (Koehn 2005) or DGT (Steinberger et al. 2013)) as the corpus and a glossary (e.g., IATE (Johnson and Macphail 2000)) or thesaurus (e.g., Eurovoc (Steinberger et al. 2002)) as the terminology gold standard list. Another strategy, used by Hazem and Morin (2017), is to collect a comparable corpus manually (i.e. scientific articles in French and English from the Elsevier[6] website) and a domain specific terminological resource (i.e. UMLS[7]) as a reference termlist. Hazem and Morin (2017) also filter out those terms from the termlist that do not appear often enough in their corpus. In other cases (e.g., Haque et al. (2014)), the datasets are not available because the papers were written as part of industrial projects and the datasets are private.

2.2.2 Code and tool availability

We have discovered that no paper has made experiment code available and only a few provide access or links to tools where the experiments were conducted. But even when links to tools are provided, reproducibility and replicability may be hindered: for example, the link provided in Ideue et al. (2011) leads to a Japanese website. Another issue is the long-term availability of resources. For example, Daille and Morin (2005) conducted their experiments in *ACABIT*, an open source terminology extraction software. However, the link given in the paper does not work anymore. From the analyzed papers, the only example of bilingual term extraction and alignment tool, which is publicly available, is the Sketch Engine term extraction module, described by Baisa et al. (2015).

None of the papers analyzed in this section fulfill the ideal scenario described at the start of this section (i.e. a single link with code and all datasets) which severly hinders any replicability attempts as will be evident from our own experiments described in this paper.

3 Replicating a machine learning approach to bilingual term alignment and reproducing its results

This section describes our efforts in replicating a machine learning approach to bilingual term alignment described in Aker et al. (2013),by which we extend our initial experiments and analysis (Repar et al. 2018). Section 3.1 describes the original approach and Sect. 3.2 contains an overview of our attempts to replicate it.

3.1 Description of the original approach

The original approach designed by Aker et al. (2013) was developed to align terminology from comparable (or parallel) corpora using machine-learning techniques. They use terms from the Eurovoc (Steinberger et al. 2002) thesaurus and train an SVM binary classifier (Joachims 2002) (with a linear kernel and the

[6] https://www.elsevier.com/.

[7] https://www.nlm.nih.gov/research/umls/.

trade-off between training error and margin parameter c = 10). The task of bilingual alignment is treated as a binary classification—each term from the source language *S* is paired with each term from the target language *T* and the classifier then decides whether the aligned pair is correct or incorrect. They then extract features (dictionary and cognate-based) to be used by the classifier. They run their experiments on the 21 official EU languages covered by Eurovoc with English always being the source language (20 language pairs altogether). They evaluate the performance on a held-out term pair list from Eurovoc using recall, precision and F-measure for all 20 languages. Next, they propose an experimental setting for a simulation of a real-world scenario where they collect English-German comparable corpora of two domains (IT, automotive) from Wikipedia, perform monolingual term extraction using the system by Pinnis et al. (2012) followed by the bilingual alignment procedure described above and manually evaluate the results (using two evaluators). They report excellent performance on the held-out term list with many language pairs reaching 100% precision and the lowest recall being 65%. For Slovenian, which is of our main interest, as well as for the additional target languages that we selected, namely French and Dutch, the reported results were excellent with perfect or nearly perfect precision and good recall for all three language pairs. The reported results of the manual evaluation phase were also good, with two evaluators agreeing that at least 81% of the extracted term pairs in the IT domain and at least 60% of the extracted term pairs in the automotive domain can be considered exact translations.

3.1.1 Features

Aker et al. (2013) use two types of features that express correspondences between the words (composing a term) in the target and source language (for a detailed description see Table 2:

- 7 dictionary-based (using Giza++) features which take advantage of dictionaries created from large parallel corpora of which 6 are direction-dependent (source-to-target or target-to-source) and 1 direction-independent—resulting in altogether 13 features, and

- 5 cognate-based (on the basis of Gaizauskas et al. (2012)) which utilize string-based word similarity between languages.

To match words with morphological differences, they do not perform direct string matching but utilize Levenshtein Distance. Two words were considered equal if the Levenshtein Distance (Levenshtein 1966) was equal or higher than 0.95. For closed-compounding languages, they check whether the compound source term has an initial prefix that matches the translation of the first target word, provided that translation is at least 5 characters long.

Table 2 Features used in the experiments

Feature	Cat	Description	Type
isFirstWordTranslated	Dict	Checks whether the first word of the source term is a translation of the first word in the target term (based on the Giza++ dictionary)	Bin
isLastWordTranslated	Dict	Checks whether the last word of the source term is a translation of the last word in the target term	Bin
percentageOfTranslatedWords	Dict	Ratio of source words that have a translation in the target term	Num
percentageOfNotTranslatedWords	Dict	Ratio of source words that do not have a translation in the target term	Num
longestTranslatedUnitInPercentage	Dict	Ratio of the longest contiguous sequence of source words which has a translation in the target term (compared to the source term length)	Num
longestNotTranslatedUnitInPercentage	Dict	Ratio of the longest contiguous sequence of source words which do not have a translation in the target term (compared to the source term length)	Num
Longest Common Subsequence Ratio (LCSSR)	Cogn	Measures the longest common non-consecutive sequence of characters between two strings (divided by the length of the longest string)	Num
Longest Common Substring Ratio (LCSTR)	Cogn	Measures the longest common consecutive string (LCST) of characters that two strings have in common (divided by the length of the longest string)	Num
Dice similarity	Cogn	2*LCST / (len(source) + len(target))	Num
Needlemann-Wunsch distance	Cogn	LCST / min(len(source), len(target))	Num
Normalized Levenshtein distance (nLD)	Cogn	1 − LD / max(len(source), len(target))	Num
isFirstWordCovered	Comb	A binary feature indicating whether the first word in the source term has a translation or transliteration in the target term	Bin
isLastWordCovered	Comb	A binary feature indicating whether the last word in the source term has a translation or transliteration in the target term	Bin
percentageOfCoverage	Comb	Returns the percentage of source term words which have a translation or transliteration in the target term	Num
percentageOfNonCoverage	Comb	Returns the percentage of source term words which have neither a translation nor transliteration in the target term	Num
difBetweenCoverageAndNonCoverage	Comb	Returns the difference between the last two features	Num

Note that some features are used more than once because they are direction-dependent

Additional features are also constructed by:

- Using language pair specific transliteration rules to create additional cognate-based features. The purpose of this task was to try to match the cognate terms while taking into account the differences in writing systems between two languages: e.g. Greek and English. Transliteration rules were created for both directions (source-to-target and target-to-source) separately and cognate-based features were constructed for both directions - resulting in additional 10 cognate-based features with transliteration rules.
- Combining the dictionary and cognate-based features in a set of combined features where the term pair alignment is correct if either the dictionary or the cognate-based method returns a positive result. This process resulted in additional 10 combined features.[8]

At the end of the feature construction phase, there were 38 features: 13 dictionary-based, 5 cognate-based, 10 cognate-based features with transliteration rules and 10 combined features.

3.1.2 Data source and experiments

Using Giza++, Aker et al. (2013) create source-to-target and target-to-source word alignment dictionaries based on the DGT translation memory (Steinberger et al. 2013). The resulting dictionary entries consist of the source word *s*, its translation *t* and the number indicating the probability that *t* is an actual translation of *s*. To improve the performance of the dictionary-based features, the following entries were removed from the dictionaries:

- entries where probability is lower then 0.05.
- entries where the source word was less than 4 characters and the target word more than 5 characters long and vice versa in order to avoid translations of stop word to content words.)

The next step is the creation of term pairs from the Eurovoc (Steinberger et al. 2002) thesaurus, which at the time consisted of 6797 terms. Each non-English language was paired with English. The test set consisted of 600 positive (correct) term pairs—taken randomly out of the total 6797 Eurovoc term pairs—and around 1.3 million negative pairs which were created by pairing each source term with 200 distinct incorrect random target terms. Aker et al. (2013) argue that this was done to simulate real-world conditions where the classifier would be faced with a larger number of negative pairs and a comparably small number of positive ones. The 600 positive term pairs were further divided into 200 pairs where both (i.e. source and target) terms were single words, 200 pairs with a single word only on one side and

[8] For combined features, a word is considered as covered if it can be found in the corresponding set of Giza++ translations or if one of the cognate-based measures (Longest Common Subsequence, Longest Common Substring, Levenshtein Distance, Needleman-Wunsch Distance, Dice) is 0.70 or higher (set experimentally by Aker et al. (2013)).

200 pairs with multiple-word terms on both sides. The remaining positive term pairs (approximately 6200) were used as training data along with additional 6200 negative pairs. These were constructed by taking the source side terms and pairing each source term with one target term (other than the correct one). Using this approach, Aker et al. (2013) achieve excellent results with 100% precision and 66% recall for Slovenian and French and 98% precision and 82% recall for Dutch.

3.2 Replication of the approach

The first step in our approach was to replicate the algorithm described by Aker et al. (2013). The initial premise is the same: given two lists of terms from the same domain in two different languages, we would like to align the terms in the two lists to get one bilingual glossary to be used in a variety of settings (computer-assisted translation, machine translation, ontology creation etc.). We followed the approach described above faithfully except in the following aspects[9]:

- Instead of the entire set of Eurovoc languages, we have initially focused only on the English-Slovenian language pair (Repar et al. 2018). In the current paper, we add two additional language pairs (English-French, English-Dutch) to see whether our findings can be generalised across different languages. We selected languages from different language families, as the importance of cognates is dependent on the similarity between languages (for example, Dutch and English (being both Germanic languages) presumably have a higher number of cognates).
- We use newer datasets. The Eurovoc thesaurus version that we used contained 7,083 terms for Slovenian[10] and 7,181 terms for French[11] and Dutch.[12] Similarly, the DGT translation memory contains additional content not yet present in 2013.[13] For English-Slovenian, we at first used the entire DGT corpus up to and including the *DGT-TM-release 2017* for deriving GIZA alignments. Later we also experimented with precomputed dictionaries by Aker et al. (2014). When performing the experiments on the other languages pairs, we did not create our own GIZA alignment, but only used the precomputed ones by Aker et al. (2014).
- Since no particular cleaning of training data (e.g., manual removal of specific entries) is described in the paper for the languages of our interest, we do not perform any.

We think that regardless of these differences, the experiments should yield similar results.

[9] Note that our original replication paper Repar et al. (2018) wrongly states that we did not utilize the compounding solution implemented by Aker et al. (2013) for addressing compouding issues in languages such as German. In fact, we did implement it and used it in all experiments.

[10] http://source.ijs.si/mmartinc/4real2018/blob/master/term_list_sl.csv.

[11] http://source.ijs.si/mmartinc/4real2018/blob/master/term_list_fr.csv.

[12] http://source.ijs.si/mmartinc/4real2018/blob/master/term_list_nl.csv.

[13] The versions of the resources used in Aker et al. (2013) were not documented or made available.

3.2.1 Problems with replicating the approach

While the general approach is clearly laid out in the article, there are several spots where further clarification would be welcome:

- There is no sufficient information about the Giza++ settings or whether the input corpora have been lemmatized. In order to improve term matching, we experimented with and without lemmatization of the Giza++ input corpora.
- There is no information about the specific character mappings rules other than a general principle of one character in the source being mapped to one or more character in the target. Since the authors cover 20 languages, it is understandable that they cannot include the actual mapping rules in the article. Therefore, we have created our own mapping rules for English-Slovenian and English-French according to the instructions in the original paper:
  - Mapping the English term to the Slovenian writing system (the character before the colon is replaced by the sequence of characters after the colon): *x:ks, y:j, w:v, q:k.*
  - Mapping the Slovenian term to the English writing system: *č:ch, š:sh, ž:zh.*
  - Mapping the French term to the English writing system: we deleted all accents e.g., é:e, ê:e.
  - Mapping the Dutch term to the English writing system: we deleted all accents and replace the digraph ij with two separate letters ij.

- Instead of the unclear Needleman–Wunsch distance formula from Aker et al. (2013) $\frac{LCST}{min[len(source)+len(target)]}$ (which implies that we should take the minimum value of the sum of the length of the target and source term) we opted for $\frac{LCST}{min[len(source),len(target)]}$ as in Nassirudin and Purwarianti (2015).
- We were not completely certain how to treat examples such as “passport—potni list”, where a single-word source term is translated by a multi-word target term and both combinations (passport—potni and passport—list) can be found in the Giza++ dictionary. In this case, our implementation returns values of 1 for both *isFirstWordTranslated* and *isLastWordTranslated* features despite the fact that the source term only has one word.
- There was a slight ambiguity on how to calculate cognate-based features: on the level of words or on the level of entire terms. We opted for the second, since the names of the cognate-based features did not imply that cognates are calculated on the word level (as was the case with the dictionary-based features) and since there was no mention in the original paper on how to combine cognate-based scores for specific word pairs in the multi-word term pairs in order to get a final cognate score for the whole term pair.
- In the original article, the *isFirstWordCovered* feature is described as “a binary feature indicating whether the first word in the source term has a translation (i.e. has a translation entry in the dictionary regardless of the score) or transliteration (i.e. if one of the cognate metric scores is above 0.7) in the target term.” While

the dictionary-based part is clear, for calculating the cognate-based feature values (e.g., of the first word in the source term), the values of the cognate metric scores concern the entire target term. As we did not find this fully intuitive, and we believe other interpretations are possible, we experimented with these settings in the adaptation of the approach (see Sect. 4.8).
To avoid ambiguities, we provide a separate document with examples of constructed features, together with the code (http://source.ijs.si/mmartinc/4real2018/blob/master/feature_examples.docx).

3.2.2 Results

The evaluation on the test set created as described in the original paper by Aker et al. (2013) shows that compared to the results reported by the authors (see line 1 in Tables 3, 4 and 5), our results are significantly worse. Despite all our efforts to follow the original approach, we were unable to match the results achieved in the original paper when running the algorithm without any changes to the original approach. When trying to follow the original paper's methodology, precision is only 3.59% and recall is 88% for the English-Slovenian language pair. The results for the other two language pairs are comparable (see line 2 in Tables 3, 4 Table 5 for details).

In Sect. 4, we provide the results of detailed analysis and additional experiments that we performed in order to reach results comparable to the original approach.

3.2.3 Attempts at establishing contact with the authors

When replicating an existing paper, especially when the code is not made available, contacting the authors for clarification (or for providing/running the code) is the most obvious step when encountering the problems or ambiguities. However, due to busy schedules of researchers, change of professional paths or other similar reasons, getting detailed help might be impossible.

This is true for our case as well. Initially, we were hopeful of getting useful feedback, as the authors already provided the software to other researchers in the past (see Arčan et al. (2014)). However, despite a friendly response, we have been able to get only a limited number of answers and many questions remained unanswered, and the auhors have not been able to share their code. We have first contacted the original authors of the paper when we were running the experiments reported in Repar et al. (2018) and did receive some answers confirming our assumptions (e.g. regarding mapping terms to the different writing systems and that the test set data was selected individually for each language pair), but several other issues remained unaddressed (in particular, what was the exact train and test data selection strategy for the EN-SL language pair). Further inquiries proved unsuccessful due to time constraints on the part of the original authors. As we expanded the paper with additional languages and experiments, we again contacted the main author, provided him the code and the paper and asked for help in

Table 3 Results on the English–Slovenian term pair

No.	Config EN-SL	Training set size	Pos/neg ratio	Precision	Recall	F-score
1	Reported by Aker et al. (2013)	12,400	1:1	1	0.6600	0.7900
2	Replicated approach	12,966	1:1	0.0359	0.8800	0.0689
3	Giza++ terms only	8306	1:1	0.0645	**0.9150**	0.1205
4	Giza++ cleaning	12,966	1:1	0.0384	0.7789	0.0731
4a	Lemmatization	12,966	1:1	0.0373	0.8150	0.0713
5	Training set 1:200	1,303,083	1:200	0.4299	0.7617	0.5496
6	Training set filtering 1	6426	1:1	0.5969	0.64167	0.6185
7	Training set filtering 2	35,343	1:10	0.9042	0.5350	**0.6723**
8	Training set filtering 3	645,813	1:200	**0.9342**	0.4966	0.6485
9	Term length filtering	6426	1:1	0.8144	0.4900	0.6119
10	Cognates approach	672,345	1:200	0.8732	0.5167	0.6492

No. 1 presents the results reported by the authors, No. 2 our replication of the approach and No. 3–10 our modifications of the first replicated approach with the aim of improving the results

Table 4 Results on the English–French language pair

No.	Config EN-FR	Training set size	Pos/neg ratio	Precision	Recall	F-score
1	Reported by Aker et al. (2013)	12,400	1:1	1	0.6600	0.7900
2	Replicated approach	13,160	1:1	0.0323	**0.8483**	0.0622
3	Giza++ terms only	8892	1:1	0.0437	0.8433	0.0830
4	Giza++ cleaning	13,160	1:1	0.0317	0.7917	0.0610
5	Training set 1:200	1,322,580	1:200	0.5273	0.6767	**0.5927**
6	Training set filtering 1	2650	1:1	0.4623	0.5517	0.5030
7	Training set filtering 2	14,575	1:10	0.9422	0.3533	0.5139
8	Training set filtering 3	266,325	1:200	**0.9791**	0.3117	0.4728
9	Term length filtering	2650	1:1	0.6791	0.3950	0.4995
10	Cognates approach	311,952	1:200	0.8603	0.3900	0.5367

No. 1 presents the results reported by the authors, No. 2 our replication of the approach and No. 3–10 our modifications of the first replicated approach with the aim of improving the results

identification of any possible mistakes leading to the results, however, we were ultimately not able to get any information which would explain the differences.

We think the original paper is generally well-written and that the main reason for occasional lack of clarity is its scope: as the authors deal with more than 20 language pairs, it would be impossible to provide specific information regarding all of them. Providing more examples would be useful, but still the code and the exact dataset are in our opinion the only way to be able to fully replicate the experiments.

Table 5 Results on the English–Dutch language pair

No.	Config EN-NL	Training set size	Pos/neg ratio	Precision	Recall	F-score
1	Reported by Aker et al. (2013)	12,400	1:1	0.9800	0.8200	0.8000
2	Replicated approach	13,160	1:1	0.0227	0.8850	0.0442
3	Giza++ terms only	7310	1:1	0.0636	**0.9317**	0.1191
4	Giza++ cleaning	13,160	1:1	0.0340	0.8500	0.0654
5	Training set 1:200	1,322,580	1:200	0.5053	0.6300	0.5608
6	Training set filtering 1	4250	1:1	0.5122	0.4917	0.5017
7	Training set filtering 2	23,375	1:10	0.6842	0.4333	0.5306
8	Training set filtering 3	427,125	1:200	**0.9356**	0.3633	0.5234
9	Term length filtering	4250	1:1	0.7621	0.3683	0.4966
10	Cognates approach	468,933	1:200	0.9101	0.5233	**0.6646**

No. 1 presents the results reported by the authors, No. 2 our replication of the approach and No. 3–10 our modifications of the first replicated approach with the aim of improving the results

4 Analysis and adaptation: experiments for improving the replicated approach

The results in our replicated experiments differ dramatically from the results obtained by Aker et al. (2013). Their approach yields excellent results with perfect or almost perfect precision and respectable recall for all three languages under our consideration.

For the EN-SL language pair, the reported results have the precision of 100% and the recall of 66%, meaning that with 600 positive term pairs in the test set, their classifier returns only around 400 positive term pairs. In contrast, in our replication attempts the classifier returned a lot of falsely classified positive term pairs. In addition to 526 true positive examples (out of a total of 600), the classifier also returns 14,194 misclassified examples—incorrect term pairs wrongly classified as correct. Similar statistics can be observed for the other two language pairs.

These results are clearly not useful for our goals which is to use the methods to continuously populate a termbase with as little manual intervention as possible. In this section we present the analysis of ambiguities in the description of the approach and the issues spotted when inspecting the results of the replicated approach, and propose several methods aiming at improving the results. To do so, we have performed experiments with regard to the following aspects:

- Giza++ terms only: using only those terms that can be found in the Giza++ training corpora (i.e. DGT).
- Giza++ cleaning.
- Lemmatization.
- Changing the ratio of positive/negative examples in the training set.
- Training set filtering.

The experiments have been initially presented for Slovenian in our short paper in the 4REAL workshop (Repar et al. 2018). Here, we provide additional analysis and extend the experiments to the other two languages under consideration. The results are reported in Sect. 4.1 to 4.5.

In the 4REAL paper, precision was already relatively high (see for example line 8 in Table 3), which is why our additional experiments focused on improving recall. We implemented several additional approaches as reported in Sect. 4.6 to 4.8:

- Removing the Needleman–Wunsch Distance feature.
- Term length filtering.
- Adding new cognate-based features.

4.1 Giza++ terms only

We thought that one of the reasons for low results can be that not all EUROVOC terms actually appear in the Giza++ training data (i.e. DGT translation memory). The terms that do not appear in the Giza++ training data could have dictionary-based features similar to the generated negative examples, which could affect the precision of a classifier that was trained on those terms. We found that only 4,153 out of 7,083 Slovenian terms of the entire EUROVOC thesaurus do in fact appear in a DGT translation memory. Using only these terms in the classifier training set did provide modest improvements of precision, recall and F-score across all three languages. For details, see line 3 in Tables 3, 4 and 5.

4.2 Giza++ cleaning

The output of the Giza++ tool contained a lot of noise and we thought it could perhaps have a detrimental effect on the results. There is no mention of any sophisticated Giza++ dictionary cleaning in the original paper beyond removing all entries where probability is lower then 0.05 and entries where the source word is less than 4 characters and the target word more than 5 characters in length and vice versa (introduced to avoid stopword-content word pairs). For clean Giza++ dictionaries, we used the resources described in Aker et al. (2014), available via the META-SHARE repository[14] (Piperidis et al. 2014), specifically, the transliteration-based approach which yielded the best results according to the cited paper.

For Slovenian and Dutch, precision and F-score improved marginally at a cost of a lower recall, while for French, precision, recall and F-score all decreased. For details, see line 4 in Tables 3, 4 and 5.

4.3 Lemmatization

The original paper does not mention lemmatization which is why we assumed that all input data (Giza++ dictionaries, EUROVOC thesaurus) is not lemmatized. They

[14] http://metashare.tilde.com, last accessed: February 14, 2019.

state that to capture words with morphological differences, they don't perform direct string matching but utilize Levenshtein Distance and two words are considered equal if the Levenshtein Distance (Levenshtein 1966) is equal or higher than 0.95. This led us to believe that no lemmatization was used. Nevertheless, we thought lemmatizing the input data could potentially improve the results which is why we adapted the algorithm to perform lemmatization (using Lemmagen (Juršič et al. 2010)) of the Giza++ input data and the EUROVOC terms. We have also removed the Levenshtein distance string matching and replaced it with direct string matching (i.e. word A is equal to word B, if word A is exactly the same as B), which drastically improved the execution time of the software.

We considered lemmatization as a factor that could explain the difference in results obtained by us and Aker et al. (2013), but our experiments on lemmatized and unlemmatized clean Giza++ dictionaries show that lemmatization does not have a significant impact on the results. Compared to the configuration with unlemmatized clean Giza++ dictionaries, in the configuration with lemmatized Giza++ dictionaries precision was slightly lower (by 0.1%), recall was a bit higher (by around 4%) and F-score was lower by 0.2%. For details, see Table 3, line 4a. As lemmatization significantly slows down the experimentation, we tested the results first on Slovenian, where the influence of the lemmatization should be the largest as it is a morphologically-rich language. As lemmatization did not improve the results, we did not repeat the experiments for French and Dutch.

4.4 Changing the ratio of positive/negative examples in the training set

In the original paper, the training set is balanced (i.e. the ratio of positive vs. negative examples is 1) but the test set is not (the ratio is around 1:2000). Since our classifier had low precision and relatively high recall, we figured that an unbalanced training set with much more negative than positive examples could improve the former. To test this, we experimented with training the classifier on unbalanced train sets with different ratios between positive and negative examples. The general tendency we noticed during experimentation is that a very unbalanced train set (ratio of 1:200 between positive and negative examples[15]) greatly improves the precision of the classifier at a cost of somewhat lower recall, when compared to balanced train set or less unbalanced train set (e.g., ratio of 1:10 between positive and negative examples). For details, see line 5 in Tables 3, 4 and 5.

4.5 Training set filtering

The original paper mentions that their classifier initially achieved low precision on Lithuanian language training set, which they were able to improve by manually removing 467 positive term pairs that had the same characteristics as negative examples from the training set. No manual removal is mentioned for Slovenian, French and Dutch.

[15] 1:200 imbalance ratio was the largest imbalance we tried, since the testing results indicated that no further gains could be achieved by further increasing the imbalance.

We have performed an error analysis and found that many incorrectly classified term pairs are cases of partial translation where one unit in a multi-word term has a correct Giza++ dictionary translation in the corresponding term in the other language. Some EN-SL examples can be seen in Table 6, and similar errors were observed for for the other two language pairs.

Based on this problem of partial translations, leading to false positive examples, we focused on the features that would eliminate these partial translations from the training set. After a systematic experimentation, we noticed that we can drastically improve precision if we only keep positive term pairs with the following feature values in the training set:

- isfirstwordTranslated = True.
- islasttwordTranslated = True.
- percentageOfCoverage > 0.66.
- isfirstwordTranslated-reversed = True.
- islasttwordTranslated-reversed = True.
- percentageOfCoverage-reversed > 0.66.

Using this approach, we managed to greatly increase precision at a cost of significant drop in recall values for all three languages. For details see line 6 (*Training set filtering 1*) in Tables 3, 4 and 5. When combining this approach with an unbalanced dataset described in the previous section, we managed to improve precision even further, but again at a cost of lower recall. For details, see lines 7 and 8 (*Training set filtering 2 and 3*) in Tables 3, 4 and 5.

4.6 Cognate feature analysis and removing the Needleman–Wunsch Distance feature

We performed an analysis of the results on the English–Slovenian language pair achieved with the best configuration for precision (line 8—*Training set filtering 3* in Table 3) in our experiments (Repar et al. 2018) and discovered that cognate term pairs were not being considered by the classifier. In a way, this was expected since in the previous step we have filtered the training set based on mostly dictionary-based features.

When analyzing the performance of the cognate-based features, we found that four (Longest Common Subsequence Ratio (LCSSR) Longest Common Substring Ratio (LCSTR), Dice Similarity (Dice), Normalized Levenshtein Distance (nLD)) out of five perform as expected with cognate term pairs having high values, but Needleman-Wunsch Distance (NWD) did not. As already mentioned in the beginning, the formula provided by the authors for computing NWD feature possibly contained an error, therefore we opted for the implementation as mentioned in Nassirudin and Purwarianti (2015). Table 7 shows the behaviour of the five cognate-based features. When we are dealing with actual cognates, all five features have high values, but when the two terms in questions are not cognates, only NWD stays high.

Table 6 Examples of negative term pairs misclassified as positive

EN	SL	Giza++
Agrarian reform	Kmetijski odpadki	Agrarian, kmetijske, 0.29737
Brussels region	Območje proste trgovine	Region, območje, 0.0970153
Energy transport	Nacionalni prevoz	Transport, prevoz, 0.442456
Fishery product	Tekstilni izdelek	Product, izdelek, 0.306948

Column 1 contains the English term, column 2 contains the Slovenian term and column 3 contains the Giza++ dictionary entry (from the non-clean version, see Sect. 4.2) responsible for positive dictionary-based features

Table 7 Cognate-based features values (showing issues with NWD)

EN	SL	LCSSR	LCSTR	Dice	nLD	NWD
hospitalisation	hospitalizacija	0.73	0.60	0.60	0.73	0.6
monopsony	monopson	0.89	0.89	0.94	0.89	1.00
fish	predstavniška demokracija	0.12	0.12	0.20	0.12	0.75
Yemen	osna obremenitev	0.25	0.25	0.38	0.25	0.80

The first two term pairs are actual cognates with all five cognate-based features having high values. The last two pairs are not cognates and show the issues with the Needleman-Wunsch Distance (NWD), which is the only measure that keeps a high value. Note that due to character mapping rules (see Section 3.2.1.), the word "predstavniška" was transformed into "predstavnishka"

For this reason, we ran our experiments without the NWD feature, but the results did not improve since the SVM classifier is known to be capable of handling noisy features.

4.7 Term length filtering

Based on error analysis, one of the major issues confusing the classifier were training examples with differing word lengths. E.g., the source term in the example would have one word, but the target term would have two. An analysis of the terms in Eurovoc for the three language pairs in question showed that 26% of the EN-SL term pairs, 34% of the EN-FR term pairs and 48% of the EN-NL term pairs have different word lengths of the source and target terms (the reason for the high ratio in EN-NL is the use of compounds in Dutch). This turned out to be one of the characteristics leading to low classification performance: for Slovenian with the replicated configuration (line 2 in Table 3) the classifier returned a total of 14,721 positively classified examples. 14,193 out of these were false positives—incorrectly aligned term pairs. A further 13376 out of these had different lengths of the source and target terms. A visual inspection of feature values indicated that there is often no clear difference between positive and negative term pairs (see Table 8).

Since this was an issue, we experimented with additional term length filtering. We took the positively classified examples from the *training set filtering 1* approach as described in Sect. 4.5 (see line 6 in the tables) and added an additional filter: if the two terms do not have the same number of words, we change the prediction from positive to negative. Using this additional filter, we achieved good precision for Slovenian (81%), and respectable for French (68%) and Dutch (76%). On the other hand, recall values were badly affected, since one third of positive term pairs in the constructed test set are terms of different word length (meaning that highest possible theoretical recall with this approach is 66%). Recall was again best for Slovenian with a value close to 50% and a bit worse for French and Dutch with a value at around 40% and 37% respectively. Consequently, F-scores were the highest for Slovenian and lower for Dutch and French. For details, see line 9 in Tables 3, 4 and 5.

From the original paper it is clear, that authors were aware of the possible complexity of terms of unequal length, as they consider terms of different lengths in the test set construction. So, we exclude the possibility that authors did not have such examples in the test set.

4.8 Cognate-based feature approach

The analysis showed that all *Training set filtering* approaches tend to overestimate the importance of Giza++ features and underestimate cognate-based features. This results in a low recall for correct cognate term pairs, which are rarely classified as positive, if their Giza++ based feature values do not show similarity with Giza++ based feature values for non-cognate correct term pairs. For example, Giza++ dictionary does not contain a Slovenian translation *pacifizem* for the English term *pacifism*, which means that the values of features *isFirstWordTranslated*, *isLastWordTranslated*, *isFirstWordTranslated-reversed* and *isLastWordTranslated-reversed* are False and the values for features *percentageOfCoverage* and *percentageOfCoverage-reversed* are zero, therefore the classifier would have a strong inclination to classify this correct term pair as incorrect, even though cognate based feature values clearly indicate that these two terms are cognates.

In order to improve the detection of cognate terms, we first propose two new cognate based features:

- isFirstWordCognate: a binary feature which returns True if the longest common consecutive string (LCST) of the first words in the source and target terms divided by the length of the longest of the two words is greater than or equal to a threshold value of 0.7 and both words are longer than 3 characters. For example, the value of the feature for the English-Slovenian term pair *Klaipeda county - Klaipedsko okrožje* would be True because the LCST for the first words in both terms is *Klaiped*, which has a length of 7. The length of the longest of the two first words in the terms (*Klaipedsko*) is 10 and 7 divided by 10 is 0.7, which is equal to the threshold value.
- isLastWordCognate: a binary feature which returns True if the longest common consecutive string (LCST) of the last words in the source and target terms

Table 8 A comparison of dictionary feature values

Source term	raw material	provision	additional resources	provision
Target term	surovine	računovodska rezervacija	surovine	urbanistični predpisi
Correctly aligned	True	True	False	False
isFirstWordTranslated	1	0	0	0
isLastWordTranslated	1	1	1	1
pctOfTransWords	0.5	1	0.5	1
pctOfNotTransWords	0.5	0	0.5	0
longestTransUnitInPct	0.5	1	0.5	1
longestNotTransUnitInPct	0.5	0	0.5	0
isFirstWordTranslated_R	0	0	0	0
isLastWordTranslated_R	1	1	1	1
pctOfTransWords_R	1	0.5	1	0.5
pctOfNotTransWords_R	0	0.5	0	0.5
longestTransUnitInPct_R	1	0.5	1	0.5
longestNotTransUnitInPct_R	0	0.5	0	0.5

The first two term pairs are correctly alligned term pairs from the training set (line 2 in Table 3), the second two are not correctly alligned term pairs. We can observe that the dictionary feature values are very similar—compare for example *raw material/surovine* and *additional resources/surovine*

divided by the length of longest of the two words is greater than or equal to a threshold value of 0.7 and both words are longer than 3 characters. For example, the value of the feature for the English-Slovenian term pair *Latin America - Latinska Amerika* would be True because the LCST for the last words in both terms is *Ameri*, which has a length of 5. The length of the longest of the two last words in the terms is 7 and 7 divided by 5 is 0.714, which is greater than the threshold value.

As having the same number of words in the source and target term could play a role in classification, we also add three new features responsible for encoding term length information:

- sourceTargetLengthMatch: a binary feature that returns True if the number of words in source and target terms match.
- sourceTermLength: returns the number of words in the source term.
- targetTermLength: returns the number of words in the target term.

Analysis of the filtered training set showed that it contained a small number of positive cognate based term pair examples, therefore the first step was to include more of them into the dataset. We build three separate datasets, each of them filtered according to the following feature values:

- isFirstWordCognate = True and isLastWordCognate = True.
- isFirstWordTranslated = True and isLastWordCognate = True.
- isFirstWordCognate = True and isLastWordTranslated = True.

The terms from these three datasets are added to the original filtered train set (we make sure that each positive term pair is represented in the new dataset only once by removing all the duplicates). The new dataset contains two distinct groups of terms, one with favorable Giza++ based features (and unfavorable cognate based features) and one with favorable cognate based features (and in some cases unfavorable Giza++ based features). Since this new dataset structure represents a classic "exclusive or" (XOR) problem which a linear classifier is unable to solve, we also replace the linear kernel of our SVM classifier with the Gaussian one.

Using this approach, precision was close to 90% (Slovenian, French) or just over 90% (Dutch), recall was just over 50% for Slovenian, around 52% for Dutch and close to 40% for French. For details, see line 10 in Tables 3, 4 and 5.

4.9 Best results

Overall, the setting with the best precision is Train set filtering 3. Compared to the replicated approach (line 2 in Tables 3, 4 and 5), it has an unbalanced dataset of 1:200 (see Section 4.4) and employs the term filtering strategy described in Sect. 4.5. However, for a small gain in recall at the price of a slight decrease in precision, a good alternative is the Cognates approach (line 10 in Tables 3, 4 and 5), which is

based on the Train set filtering 3 approach and additionally includes the cognate detection strategies described in Sect. 4.8.

5 Manual evaluation

The first part of this section contains the manual evaluation replicated from Aker et al. (2013), already reported in Repar et al. (2018), while the second part is novel and contains an evaluation using a new dataset and has a specific focus on cognate term pairs.

5.1 Replicating the manual evaluation experiments from the original paper

Similar to the original paper, we also performed manual evaluation. We selected a random subset of term pairs classified as positive by the classifier (using the *Training set filtering 3* configuration (line 8 in Table 3) that yielded the best precision). While the authors of the original approach extract monolingual terms using the term extraction and tagging tool TWSC (Pinnis et al. 2012), we use a workflow for monolingual term extraction by Pollak et al. (2012). Both use a similar approach - terms are first extracted using morphosyntactic patterns and then filtered using statistical measures: TWSC uses pointwise mutual information and TF*IDF, while Pollak et al. (2012) is based on an approach by Vintar (2010) and compares the relative frequencies of words composing a term in the domain-specific (i.e. the one we are extracting terminology from) corpus and a general language corpus.

In contrast to the original paper where they extracted terms from domain-specific Wikipedia articles (for the English-German language pair), we are using two translation memories—one containing finance-related content, the other containing IT content. Another difference is that extraction in the original paper was done on comparable corpora, but we extracted terms from parallel corpora - which is why we expected our results to be better. Each source term is paired with each target term (just as in the original paper - if both term lists contained 100 terms, we would have 10,000 term pairs) and extract the features for each term pair. The term pairs were then presented to the classifier that labeled them as correct or incorrect term translations. Afterwards, we took a random subset of 200 term pairs classified as correct and showed them to an experienced translator[16] fluent in both languages who evaluated them according to the criteria set out in the original paper:

- **1—Equivalence**: The terms are exact translations/transliterations of each other (e.g., *type—tip*).
- **2—Inclusion**: Not an exact translation/transliteration, but an exact translation/transliteration of one term is entirely contained within the term in the other language (e.g., *end date—datum*).
- **3—Overlap**: Not category 1 or 2, but the terms share at least one translated/transliterated word (e.g., *user id—uporabniško ime*).

[16] The original paper used two annotators, hence two lines for each domain in Table 4.

– **4—Unrelated**: No word in either term is a translation/transliteration of a word in the other (e.g., *level—uporabnik*[17]).

The results of the manual evaluation can be found in Table 9. Manual evaluation showed that 72% of positive term pairs in the Finance domain, and 79% of positive term pairs in the IT domain were correctly classified by the classifier. The differences between the *Finance* and *IT* datasets can be partially explained by the *Finance* dataset containing more MWE terms than the *IT* dataset (84 vs. 51 for SL and 78 vs. 49 for EN). On the one hand, this means that the chances of aligning a single word term in one language with a multi word term in another language is greater, hence the greater number of partial translations in *Finance* (category 2 - Inclusion), while on the other, single word terms means less characters for the algorithm to work with, hence the greater number of outright mistakes in *IT* (category 4 - Unrelated). Compared to the original paper, we believe these results are comparable when taking into account the different monolingual extraction procedures , the different language pairs and the human factor related to different annotators.

5.2 Evaluation on a Karst terminology gold-standard

As mentioned in Sect. 4, the best configuration in terms of precision used in Repar et al. (2018) (line 8 in Tables 3, 4 and 5) overestimates dictionary-based and underestimates cognate-based features. To alleviate this, we added additional features and filtering strategies to our approach to try to improve cognate term pair alignment (see lines 9 and 10 in the results tables). However, evaluating its performance on EUROVOC is difficult as many terms have favorable dictionary-based features due to the fact that both the Giza++ dictionary and EUROVOC are made from the same content (i.e. EU documentation). For the evaluation in this section, we therefore selected a domain, with a content type which is unlikely to be found in DGT (Steinberger et al. 2013), i.e. karstology, which is the science in the field of geomorphology, specializing in the study of karst formations.

To evaluate our bilingual term alignment approach, we used a gold standard of EN-SL aligned karst terminology,[18] which was manually created by the authors of the karstology corpus (Vintar and Grčić-Simeunović 2016). The gold standard consists of 52 English-Slovenian term pairs. For the evaluation experiment, we aligned all Slovenian term with all English terms, resulting in a dataset of 52 positive examples and 2652 negative examples. With the best configuration for precision (line 8 in Table 3), selected also as the best configuration in Repar et al. (2018), precision was 100%, but recall was only 40.4%. Many term pairs containing cognates such as “eogenetic cave—eogenetska jama”, “epigenic aquifer—epigeni vodonosnik” or “karst polje—kraško polje”, were not aligned. With the final cognate approach (line 10 in Table 3), we managed to retain 100% precision and raise the recall to 50% by finding 7 additional cognate term pairs (*aggressive*

[17] “uporabnik” means “user”.

[18] http://source.ijs.si/mmartinc/4real2018/tree/master/datasets/karst_corpus.

water—agresivna voda, *eogenetic cave—eogenetska jama*, *precipitation—precipitacija*, *ponor cave—ponorna jama*, *epigenic aquifier—epigeni vodonosnik*, *karst polje—kraško polje*, *linear stream cave—linearna epifreatična jama*). However one half of correct term pairs remain undiscovered. We believe this is due to 1) domain-specific words which are not cognates and are missing from the Giza++ dictionary (e.g., *porous aquifer—medzrnski vodonosnik* and *denuded cave—brezstropa jama*), and 2) valid cognate words which do not meet the threshold described in Sect. 4.8 (e.g. *oxidization—oksidacija*, *percolation—perkolacija* and *liquefication—likvifakcija*).[19]

6 Replicability and reproducibility of our own terminology alignment results

As mentioned before, availability of the source code can drastically improve the reproducibility of experiments, since very detailed descriptions of procedures used in the experiments are beyond the scope of most papers because of length limitations and negative effects on the readability of the paper. Since we wanted to ensure the full reproducibility of our approach, we decided to publish the source code for all the conducted experiments and results that are published in the paper. As we were aware that just the presence of source code itself does not guarantee complete reproducibility, we decided that the published code should comply to the following three criteria:

- Instructions on how to use the code should be as unambiguous, simple and clear as possible.
- Code should be bug free and running it according to the instructions should yield the exact same results as published in the paper.
- Running the code should require as little time and technical skills as possible.

In order to validate that the published code complies to these criteria, we asked three students[20] to try to reproduce the results published in the paper (Repar et al. 2018) and after that answer the following questions related to the proposed criteria:

- Did you manage to reproduce the results?
- If not, what do you think was the main problem?
- If yes, how much time did you need for replicating the experiment?
- Were the instructions clear?
- Did you run into any specific problems during any part of the replicability attempt? If yes, please describe it.
- Do you have any suggestions on how to further improve the reproducibility of the results?

[19] It might also make sense to include morphological information as a feature of the machine learning algorithm, since all these word have endings typical of cognates in their respective languages.

[20] 2 Master students (one in Economy and one in Computer science) and 1 first year PhD student in ICT.

Table 9 Manual evaluation results

Domain	1	2	3	4
Reported in Aker et al. (2013)				
IT, Ann. 1	0.81	0.06	0.06	0.07
IT, Ann. 2	0.83	0.07	0.07	0.03
Auto, Ann. 1	0.66	0.12	0.16	0.06
Auto, Ann. 2	0.60	0.15	0.16	0.09
Replication				
Finance	0.72	0.09	0.12	0.07
IT	0.79	0.01	0.09	0.12

Ann. stands for "Annotator" since the original paper uses two annotators

We also imposed a time limit of 8 hours (one working day) for the entire replicability attempt. If that limit was reached, the replicability attempt would count as unsuccessful.

The feedback we got was interesting and made us reconsider the initial source code criteria. Two out of three students managed to reproduce all the published results in less than an hour without any major problems. They did however point out some mistakes and ambiguities in the instructions on how to run the code. These were mostly connected with the programming environment used by the students, one of them using PyPI Python package manager for acquiring dependencies while the other one used the Conda environment, for which the usage instructions were not published.

The third student managed to reproduce the results in about two hours and reported some major problems with dependencies installation. He was the only person trying to reproduce the experiments in the Windows environment while the other two students used a Linux operating system, and he reported problems with the Python implementation of the Lemmagen lemmatizer (Juršič et al. 2010), which he was unable to install properly on the Windows platform. He managed to overcome the problem by manually removing the dependency from the code, by which he limited the flexibility of the published source code (he could only use it for the classification on the pre-generated train and test sets) but did not make the reproduction impossible.

While he was successful at reproducing the results for eight out of nine experiments published in the paper, he also reported a slight deviation (by less than 0.05 percentage point) from the reported recall and F-score in one of the experiments. Although we are not sure what is the exact reason for this deviation, we suspect it could be connected to the difference in operating systems.

These experiments show that programming environment and the choice of the operating system can have an unexpected negative impact on the reproducibility. While attaching code usage instructions for every possible programming environment and operating system is practically impossible, we do believe that the results of this experiment show that a published source code should comply to one additional criteria:

– Instructions should clearly specify on which operating system and in which programming environment the reported results were produced.

We have updated the usage instructions for our source code to comply with these criteria.

7 Reusability of our code in the ClowdFlows online platform

Because we want to make sure that our terminology alignment system is also available to a wider audience of users with lower level of technical skills (e.g., translators or linguists) and because we want to encourage a very simple reusability of our system, we have implemented the system into a cloud-based visual programming platform ClowdFlows (Kranjc et al. 2012). The ClowdFlows platform employs a visual programming paradigm in order to simplify the representation of complex data mining procedures into visual arrangements of their building blocks. Its graphical user interface is designed to enable the users to connect processing components (i.e. widgets) into executable pipelines (i.e. workflows) on a design canvas by a drag and drop technique, reducing the complexity of composition and execution of these workflows. The platform also enables online sharing of the composed workflows.

We took pretrained models of our terminology alignment system for English-Slovenian, English-French and English-Dutch alignment and packed them in a widget *Terminology alignment*, so it can be used out-of-the-box. The widget takes two columns of the Pandas dataframe (McKinney 2011) containing the source and target terms as inputs and returns a dataframe containing aligned term pairs. The user needs to define the names of the columns in the dataframe containing source and target language termlists, and the language of alignment as parameters. The user can also switch between configurations *Training set filtering 3* with the best precision and *Cognates approach* with the on average best F-score for all three languages while still having good precision by either enabling or disabling the *Maximize recall* widget parameter. Such an end to end system for bilingual terminology alignment in ClowdFlows is implemented at: http://clowdflows.org/workflow/13789/.[21] Another widget called *Terminology alignment evaluation* is used for determining the performance of the system (if we have a gold standard available), taking as input the dataframe produced by the *Terminology alignment* widget and a dataframe containing true alignments, and outputting the performance score in terms of precision, recall and F-score.

Workflow in Fig. 1 (available at http://clowdflows.org/workflow/13753/) is a ClowdFlows implementation for terminology alignment and evaluation. The source and target terminologies are both loaded from a CSV file with the help of the *Load Corpus From CSV* widget and fed as input to the *Terminology alignment* widget,

[21] Note that the execution time of term alignment increases rapidly with the increase in number of terms, e.g., alignment of hundred terms takes around five minutes, while it takes about one hour for alignment of thousand terms.

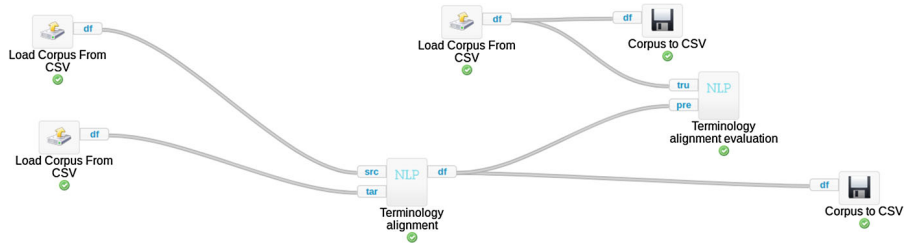

Fig. 1 ClowdFlows implementation of the system for terminology alignment and evaluation available at http://clowdflows.org/workflow/13753/

which returns a dataframe with alignments. These are written to a CSV file with the *Corpus to CSV* widget and also fed to the *Terminology alignment evaluation* widget together with the dataframe containing true alignments (which was also loaded from a CSV file with the *Load Corpus From CSV* widget) in order to estimate the performance of the system. In addition, term alignment widget can also be incorporated into a bilingual terminology extraction workflow (Pollak et al. 2012). The workflow with the newly added term alignment widget, is available at http://clowdflows.org/workflow/13723/), where a user can now input text from a specific domain in Slovenian and English and get aligned terminology as output.

8 Conclusions and future work

Based on our research and attempts at replicating a bilingual terminology alignment paper reproducing its results, we propose a set of best practices any bilingual terminology extraction paper (and more generally every NLP paper) should fulfill to facilitate reproducibility and replicability of the experiments:

- *Dataset availability.* Availability of datasets (i.e. gold standard term lists, corpora) is an essential prerequisite for successful replication.
- *Experiment code availability.* The main task of reproducibility and replicability experiments is often to reconstruct the experiments in computer code. It is a cumbersome process which inevitably requires that the reproducer/replicator makes educated guesses at some point since a detailed description of the code is beyond the scope of most papers. Having the original code available greatly increases the ease of reproducibility and replicability experiments.
- *Tool availability.* Availability of a tool or application (online or offline) where experiments can be conducted eases reproducibility and replicability, but also enables the reusability of results by a larger community.
- Finally, releasing intermediate results, configuration settings and the actual outcomes of individual experiments, while not essential, would provide future researchers with an even greater possibility of successful reproduction of the paper's results.

A prerequisite for successful reproduction and replication is a clearly written research paper. However as is evident from our example, it is often difficult to include all necessary implementation notes given the length restrictions of the paper. For this reason, another best practice would be to provide relevant implementation examples alongside the code (which is what we did for feature construction.[22]) Finally, as the experiment in Sect. 6 showed, even code itself is sometimes not enough without additional implementation notes and information on the operating systems and software used. In addition, testing the code by non-authors is strongly recommended.

Our attempts focused on the approach to bilingual term alignment using machine learning by Aker et al. (2013). They approach term alignment as a bilingual classification task—for each term pair, they create various features based on word dictionaries (i.e. created with Giza++ from the DGT translation memory) and word similarities across languages. They evaluated their classifier on a held-out set of term pairs and additionally by manual evaluation. Their results on the held-out set were excellent, with 100% precision and 66% recall for the English-Slovenian and English-French language pair and 98% precision and 82% recall for English-Dutch.

Our reproduction attempt focused on three language pairs: English-Slovenian, English-Dutch and English-French (in contrast with the original article where they had altogether 20 language pairs) and we were unable to reproduce the results following the procedures described in the paper. In fact, our results have been dramatically different from the original paper with precision being less than 4% and recall close to 90% for all three language pairs under consideration. We then tested several different strategies for improving the results ranging from Giza++ dictionary cleaning, lemmatization, different ratios of positive and negative examples in the training and test sets, training set filtering based on feature values and term length, and adding new cognate-based features. The most effective strategies employed unbalanced training set and training set filtering based on certain feature values which resulted in precision exceeding 90% for all three language combinations (*Training set filtering 3* configuration, line 8 in Tables 3, 4 and 5). It is possible that in the original experiments authors performed a similar training set filtering strategy, because the original paper mentions that their classifier initially achieved low precision on Lithuanian language training set, which they were able to improve by manually removing positive term pairs that had the same characteristics as negative examples from the training set. However, no manual removal is mentioned for Slovenian, Dutch or French. Further attempts were directed at boosting recall and the performance of cognate-based features. By adding additional cognate-based features, we were able to improve recall by around 16% for Dutch, 8% for French and by around 2% for Slovenian (over the *Training set filtering 3* configuration) at a cost of a moderate drop in precision.

For evaluation we focused only on Slovenian, which is our native language and of primarily interest for our applied tasks. We performed manual evaluation similar to the original paper and reached roughly the same results with our adapted approach. In addition, because we discovered that Eurovoc data is of limited use for

[22] http://source.ijs.si/mmartinc/4real2018/blob/master/feature_examples.docx.

evaluating the performance of cognate-based features, we ran experiments on an English-Slovenian karstology gold standard term list. With the *Cognates approach* configuration (line 10 in Tables 3, 4 and 5), we improved recall by 11% (compared to the *Training set filtering 3* configuration) and a qualititive analysis of the results showed that the new strategies for boosting the performance of cognate-based features do indeed result in more cognate term pairs being properly aligned.

This paper demonstrates some of the obstacles for research reproducibility and replicability, with the prime one being code unavailability. Had we had access to the code of the original experiments, it is highly likely that replicating the original paper would be a trivial matter. Also in this particular case, the discrepancy in the results could be attributed to the scope of the original paper - with more than 20 languages—which is also a demonstration of very impressive approach—it would be impossible to describe procedures for all of them. We weren't able to reproduce the results of the original paper, but after developing the optimization approaches described above over the course of several months, we were able to reach a useful outcome at the end. We believe that providing supplementary material online, i.e. the code and datasets, is the only way of assuring complete reproducibility of results. For this reason, in order to help with any future reproducibility/replicability attempts of our paper, we are publishing the code at: http://source.ijs.si/mmartinc/4real2018.

In terms of future work, we plan to expand the feature set by introducing the features derived from the distributions in parallel corpora (e.g. co-frequency, logDice and other measures, see Baisa et al. (2015)), as well as investigate novel methods using cross-lingual embeddings. In terms of reproducibility, we plan to extend the study to a systematic comparison of different term alignment and term extraction methods.

Acknowledgements This paper is supported by European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The authors acknowledge also the financial support from the Slovenian Research Agency for research core funding for the programme Knowledge Technologies (No. P2-0103). The authors also acknowledge the project TermFrame—Terminology and Knowledge Frames across Languages (No. J6-9372), which was financially supported by the Slovenian Research Agency. We would also like to thank the company Iolar, for allowing us to use the data from the translation memories in one of the experiments.

References

Aker, A., Paramita, M., & Gaizauskas, R. (2013). Extracting bilingual terminologies from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: Vol. 1*. Long Papers (pp 402–411).

Aker, A., Paramita, M. L., Pinnis, M., & Gaizauskas, R. (2014). Bilingual dictionaries for all EU languages. In *Proceedings of 9th International Conference on Language Resources and Evaluation*. (pp 2839–2845).

Arčan, M., Turchi, M., Tonelli, S., & Buitelaar, P. (2014). Enhancing statistical machine translation with bilingual terminology in a CAT environment. https://doi.org/10.13140/2.1.1019.8404.

Bader, B. W., & Chew. P. A. (2008). Enhancing multilingual latent semantic analysis with term alignment information. In *Proceedings of the 22nd International Conference on Computational Linguistics: Vol. 1. Association for Computational Linguistics* (pp 49–56).

Baisa, V., Ulipová, B., & Cukr, M. (2015). Bilingual terminology extraction in Sketch Engine. In *9th Workshop on Recent Advances in Slavonic Natural Language Processing*. (pp 61–67).

Baldwin, T., & Tanaka, T. (2004). Translation by machine of complex nominals: Getting it right. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*. (pp 24–31).

Bouamor, D., Semmar, N., Zweigenbaum, P. (2013). Context vector disambiguation for bilingual lexicon extraction from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: Vol. 2: Short Papers*. (pp 759–764).

Branco, A., Calzolari, N., & Choukri, K. (eds) (2018). 4REAL 2018—Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language, ELRA.

Branco, A., Cohen, K. B., Vossen, P., Ide, N., & Calzolari, N. (2017). Replicability and reproducibility of research results for human language technology: introducing an LRE special section.

Camerer, C. F., Dreber, A., Forsell, E., Ho, T. H., Huber, J., Johannesson, M., et al. (2016). Evaluating replicability of laboratory experiments in economics. *Science*, *351*(6280), 1433–1436.

Cao, Y,, & Li, H. (2002). Base noun phrase translation using web data and the EM algorithm. In *Proceedings of the 19th International Conference on Computational Linguistics: Vol. 1*. (pp 1–7).

Chiao, Y. C., & Zweigenbaum, P. (2002). Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of the 19th International Conference on Computational Linguistics: Vol. 2*. (pp 1–5).

Cohen, K. B., Xia, J., Zweigenbaum, P., Callahan, T., Goss, F., Ide, N., Névéol, A., Grouin, C., & Hunter, L. E. (2018). Three dimensions of reproducibility in natural language processing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. (pp 156–165).

Daille, B., Gaussier, E., & Langé, J. M. (1994). Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th Conference on Computational Linguistics: Vol. 1*. (pp 515–521).

Daille, B., & Morin, E. (2005). French-English terminology extraction from comparable corpora. *Natural Language Processing - IJCNLP*, *2005*, 707–718.

Fokkens, A., Van Erp, M., Postma, M., Pedersen, T., Vossen, P., & Freire, N. (2013). Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: Vol. 1: Long Papers*. (pp 1691–1701).

Foo, J. (2012). Computational terminology: Exploring bilingual and monolingual term extraction. PhD thesis, Linköping University Electronic Press.

Fung, P., & Yee, L. Y. (1998). An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th International Conference on Computational Linguistics: Vol. 1*. (pp 414–420).

Gaizauskas, R., Aker, A., & Yang Feng, R. (2012). Automatic bilingual phrase extraction from comparable corpora. In *24th International Conference on Computational Linguistics*. (pp 23–32).

Gale, W., & Church, K. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, *19*(1), 75–102.

Gaussier, E. (1998). Flow network models for word alignment and terminology extraction from bilingual corpora. In *Proceedings of the 17th International Conference on Computational Linguistics: Vol. 1*. (pp 444–450).

Ha, L. A., Fern, G., Mitkov, R., Corpas, G. (2008). Mutual bilingual terminology extraction. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. (pp 1818–1824).

Haque, R., Penkale, S., & Way, A. (2014). Bilingual termbank creation via log-likelihood comparison and phrase-based statistical machine translation. In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*. (pp 42–51).

Hazem, A., & Morin, E. (2016). Efficient data selection for bilingual terminology extraction from comparable corpora. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. (pp 3401–3411).

Hazem, A., & Morin, E. (2017). Bilingual word embeddings for bilingual terminology extraction from specialized comparable corpora. In *Proceedings of the 8th International Joint Conference on Natural Language Processing: Vol. 1: Long Papers*. (pp 685–693).

Ideue, M., Yamamoto, K., Utiyama, M., & Sumita., E. (2011). A comparison of unsupervised bilingual term extraction methods using phrase tables. In *Proceedings of the 13th Machine Translation Summit*. (pp 346–351).

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms*. Alphen aan den Rijn: Kluwer Academic Publishers.

Johnson, I., & Macphail, A. (2000). IATE–Inter-Agency Terminology Exchange: Development of a single central terminology database for the institutions and agencies of the European Union. In *Proceedings of the Workshop on Terminology resources and computation, LREC 2000 Conference*.

Juršič, M., Mozetič, I., Erjavec, T., & Lavrač, N. (2010). Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, *16*(9), 1190–1214.

Kano, Y., Baumgartner, W. A, Jr., McCrohon, L., Ananiadou, S., Cohen, K. B., Hunter, L., et al. (2009). U-compare: Share and compare text mining tools with uima. *Bioinformatics*, *25*(15), 1997–1998.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit: Vol. 5*. (pp 79–86).

Kontonatsios, G., Korkontzelos, I., Tsujii, J., & Ananiadou, S. (2014). Combining string and context similarity for bilingual term alignment from comparable corpora. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (pp 1701–1712).

Kranjc, J., Podpečan, V., & Lavrač, N. (2012). Clowdflows: A cloud based scientific workflow platform. In *Proceedings of Machine Learning and Knowledge Discovery in Databases, ECML/PKDD (2)*. Springer. (pp 816–819).

Kupiec, J. (1993). An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*. (pp 17–22).

Lee, L., Aw, A., Zhang, M., & Li, H. (2010). Em-based hybrid model for bilingual terminology extraction from comparable corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics*. (pp 639–646).

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, *10*, 707.

Macken, L., Lefever, E., & Hoste, V. (2013). Texsis: Bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology International Journal of Theoretical and Applied Issues in Specialized Communication*, *19*(1), 1–30.

McKinney, W. (2011). Pandas: A foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing. (pp 1–9).

Morin, E., Daille, B., Takeuchi, K., Kageura, K. (2007). Bilingual terminology mining-using brain, not brawn comparable corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. (pp 664–671).

Morin, E., Daille, B., Takeuchi, K., & Kageura, K. (2008). Brains, not brawn: The use of smart comparable corpora in bilingual terminology mining. *ACM Transactions on Speech and Language Processing*, *7*(1), 1.

Nassirudin, M., & Purwarianti, A. (2015). Indonesian-Japanese term extraction from bilingual corpora using machine learning. In *International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2015*. (pp 111–116).

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*(1), 19–51.

Pedersen, T. (2008). Empiricism is not a matter of faith. *Computational Linguistics*, *34*(3), 465–470.

Pinnis, M., Ljubešić, N., Stefanescu, D., Skadina, I., Tadić, M., & Gornostaya, T. (2012). Term extraction, tagging, and mapping tools for under-resourced languages. In *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012)*. (pp 20–21).

Piperidis, S., Papageorgiou, H., Spurk, C., Rehm, G., Choukri, K., Hamon, O., Calzolari, N., Del Gratta, R., Magnini, B., & Girardi, C. (2014). Meta-share: One year after. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*. (pp 1532–1538).

Pollak, S., Vavpetič, A., Kranjc, J., Lavrač, N., & Vintar V,. (2012). NLP workflow for on-line definition extraction from English and Slovene text corpora. *11th Conference on Natural Language Processing, KONVENS 2012 - Empirical Methods in Natural Language Processing* (pp. 53–60). Vienna: Austria.

Prinz, F., Schlange, T., & Asadullah, K. (2011). Believe it or not: How much can we rely on published data on potential drug targets? *Nature Reviews Drug Discovery*, *10*(9), 712.

Rapp, R. (1999). Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. (pp 519–526).

Repar, A., Martinc, M., & Pollak, S. (2018). Machine learning approach to bilingual terminology alignment: Reimplementation and adaptation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., Schlüter, P. (2013). DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*.

Steinberger, R., Pouliquen, B., & Hagman, J. (2002). Cross-lingual document similarity calculation using the multilingual thesaurus eurovoc. Computational Linguistics and Intelligent Text Processing. (pp 101–121).

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In: Chair) NCC, Choukri K, Declerck T, Dogan MU, Maegaard B, Mariani J, Odijk J, Piperidis S (eds) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey.

Vintar, Š., & Grčić-Simeunović (2016). Definition frames as language-dependent models of knowledge transfer. Fachsprache : internationale Zeitschrift für Fachsprachenforschung, - didaktik und Terminologie. (pp 43–58).

Vintar, Š. (2010). Bilingual term recognition revisited: The bag-of-equivalents term alignment approach and its evaluation. *Terminology International Journal of Theoretical and Applied Issues in Specialized Communication*, *16*(2), 141–158.

Wieling, M., Rawee, J., & van Noord, G. (2018). Reproducibility in computational linguistics: Are we willing to share? *Computational Linguistics*, *44*(4), 641–649.

Yentis, S., Campbell, F., & Lerman, J. (1993). Publication of abstracts presented at anaesthesia meetings. *Canadian Journal of Anaesthesia*, *40*(7), 632–634.

TermEnsembler

An ensemble learning approach to bilingual term extraction and alignment

Andraž Repar, Vid Podpečan, Anže Vavpetič, Nada Lavrač & Senja Pollak
Jožef Stefan Institute

This paper describes TermEnsembler, a bilingual term extraction and alignment system utilizing a novel ensemble learning approach to bilingual term alignment. In the proposed system, the processing starts with monolingual term extraction from a language industry standard file type containing aligned English and Slovenian texts. The two separate term lists are then automatically aligned using an ensemble of seven bilingual alignment methods, which are first executed separately and then merged using the weights learned with an evolutionary algorithm. In the experiments, the weights were learned on one domain and tested on two other domains. When evaluated on the top 400 aligned term pairs, the precision of term alignment is over 96%, while the number of correctly aligned multi-word unit terms exceeds 30% when evaluated on the top 400 term pairs.

Keywords: bilingual terminology alignment, terminology extraction, ensemble learning, evolutionary algorithm

1. Introduction

With the onset of globalized markets, the need for effective multilingual communication has never been greater. Language industry, a term used to describe collectively the companies that offer translation and other related language services, has been steadily growing for several years and the increase in the volume of translated words brought along the need to streamline the translation process with automated solutions. In the 1990s, translation companies embraced computer-assisted translation (CAT) tools that allow them to store translations in a database and recycle them in future translation tasks.

https://doi.org/10.1075/term.00029.rep
Terminology 25:1 (2019), pp. 93–120. issn 0929-9971 | e-issn 1569-9994

Parallel to this process, another distinct (but related) development took place which revolved around terminology in the translation process. While several solutions and tools have been proposed, terminology remains one of the main problem areas for the translation industry. For example, a 2014 report[1] by SDL, a market leader in translation and terminology management software solutions, showed that among 140 companies, 51 percent of the respondents did not have a terminology management process in place, while a survey by Schmitz and Straub (2016) showed that among 800 respondents, 89.5 percent often or constantly experience that different organizational areas or employees use different terms for the same concept and that 51.9 percent of employees often or constantly cannot understand terms immediately. SDL Translation Technology Insights Series survey,[2] which focused on translation quality conducted among a mix of translation buyers, language service providers and freelance translators, found that "inconsistencies in the use of terminology" is the number one reason of translation rework (i.e. when translation is deemed not good enough and the source text has to be translated again) and recommended that, in order to improve translation quality, terminology management be prioritized.

Due to the early adoption of CAT tool technology in the translation industry, most translation companies have large repositories of translation memories. To illustrate, Gouadec (2007) reported that among more than 430 translation job advertisements surveyed, 95 percent contain a requirement for a "translation memory skill." In the period since that study, translation memories have remained a central component of any translation company business model.

This paper addresses the above-mentioned needs of the translation industry by proposing a system for semi-automated terminology extraction and alignment, currently focusing on English and Slovenian. The system, developed for one of the largest language service providers in Southeast Europe, consists of:

- A concept-oriented terminology database, where all the data is stored, allowing import from and export into industry-standard terminology management formats.
- A terminology extraction workflow, including automated extraction or import of manually defined monolingual terminology, followed by a novel approach to term alignment utilizing an evolutionary algorithm to combine the results of several individual bilingual term alignment methods.

1. SDL Research – Terminology: An End-to-End Perspective (http://www.sdl.com/download/terminology-an-endtoend-perspective/71114/). Accessed 3 March 2017.

2. Research Study 2016: Translation Technology Insights – Productivity (https://www.sdl.com/download/tti16-productivity/109572/). Accessed 3 March 2017.

– A web interface for managing the database and controlling the extraction and alignment algorithms.
– Additional functionalities for extraction of good example sentences and identification of the domain in which the term is used.

The novel approach to bilingual term alignment is the main contribution of this work. We systematically compare several existing term alignment methods, propose a novel Phrase-Table-Based Alignment (PTBA) method based on Pialign (Neubig et al. 2011), as well as a novel methodology using an evolutionary algorithm to combine solutions of an ensemble of elementary term alignment algorithms. We evaluate the performance of the system on three different domains, where one domain was used for training and two domains were used for testing the proposed approach.

This paper is structured as follows: Section 2 describes the related work, Section 3 describes the system and its methodology, Section 4 contains the experiments and results, while Section 5 contains the conclusions and plans for future work.

2. Related work

Terminology extraction refers to structuring terminological knowledge from unstructured text. Parallel translation databases (i.e. translation memories), which are omnipresent in the translation industry, lend themselves nicely to automated terminology extraction. In addition to terminology, various other types of information can be extracted, such as named entities, collocations or good examples.

In terms of input text, we can distinguish between monolingual terminology extraction, where terms are extracted from text in one language, and bilingual or multilingual terminology extraction, where the goal is to extract and align terms from text in two or more languages. A brief survey of related work is presented in Sections 2.1 and 2.2, respectively.

2.1 Monolingual term extraction

In the broadest sense, there are two different approaches to monolingual term extraction: linguistic and statistical. The linguistic approach utilizes the distinctive linguistic aspects of terms – most often their syntactic patterns, while the statistical approach takes advantage of term frequencies in the corpus. However, most state-of-the-art systems are hybrid, using a combination of the two approaches; e.g., Justeson and Katz (1995) first define part-of-speech patterns of terms and then use simple frequencies to filter the term candidates.

Many terminology extraction algorithms are based on the concepts of termhood and unithood defined by Kageura and Umino (1996). Termhood is "the degree to which a stable lexical unit is related to some domain-specific concepts" and unithood is "the degree of strength or stability of syntagmatic combinations and collocations." Termhood-based statistical measures function on a presumption that a term's relative frequency will be higher in domain-specific corpora than in the general language. Several approaches utilizing termhood have been developed, including those by Ahmad et al. (2000) and Vintar (2010). Common statistical measures are used to measure unithood, such as mutual information (Daille et al. 1994) or t-test (Wermter and Hahn 2005).

In the last few years, word embeddings – vectors of real numbers representing words on a corpus – have become a very popular natural language processing technique. The turning point was the paper by Mikolov et al. (2013) describing word2vec, a word embedding toolkit that can create vector space models much faster than previous attempts. Several attempts have already been made to utilize word embeddings for terminology extraction (e.g. Amjadian et al. (2016), Wang et al. (2016), Khan et al. (2016) and Zhang et al. (2018)).

2.2 Bilingual term extraction and alignment

At the highest level, bilingual terminology extraction can be divided into extraction from comparable and extraction from parallel corpora, where parallel corpora are composed of source texts and their translations in one or more different languages, while comparable corpora are composed of monolingual texts collected from different languages using similar sampling techniques (McEnery et al. 2006). For alignment of terms between the two languages, the methods typically utilize the idea that a term and its translation tend to occur in similar lexical contexts (Daille and Morin 2005).

In the language-industry context, taking into account parallel bilingual sentence pairs, stored in the translation memory, brings significant advantages to the task of terminology extraction. Broadly speaking, there are two distinct approaches to bilingual terminology extraction from parallel corpora according to Foo (2012):

- Align-extract, where we first align single and multi-word units in parallel sentences and then extract the relevant terminology from a list of candidate term pairs, and
- Extract-align, where we first extract monolingual candidate terms from both sides of the corpus and then align the terms.

A state-of-the-art align-extract approach is proposed by Macken et al. (2013) utilizing a chunk-based alignment method to produce a list of candidate term pairs, which are then filtered using statistical methods.

The extract-align approach is the more common of the two. Kupiec (1993) describes an algorithm for noun phrase extraction followed by alignment with a statistical estimation algorithm, achieving precision of 90 percent on the highest ranking candidate pairs. Vintar (2010) describes an extract-align approach named "bag-of-equivalents", where after monolingual extraction, the term pairs are aligned with the help of word alignment probabilities. Baisa et al. (2015) describe a frequency-based term alignment algorithm utilizing a variation of logDice to score the strength of the candidate term pair alignment. Haque et al. (2014) first generate candidate terms monolingually and then build a phrase table using the Moses toolkit (Koehn et al. 2007) and compare the extracted terms with the phrases in the table. Precision among the top 100 candidate term pairs often exceeds 90 percent. Aker et al. (2013) treat bilingual term alignment as a binary classification task, achieving good results. More recently, Hazem and Morin (2017) experiment with word embeddings used to augment bilingual terminology extraction from specialized comparable corpora (achieving precision of 70.9 percent).

The approach proposed in this paper is based on the idea of utilizing evolutionary algorithms which mimic biological evolution (i.e. reproduction, mutation, selection) to optimize the stated objective. Specifically, we use the genetic algorithm implementation in DEAP (*Distributed Evolutionary Algorithms in Python*) by Fortin et al. (2012) to build a term alignment ensemble.

3. TermEnsembler system and methodology

In this section, we describe the functionality of the developed TermEnsembler system, starting with the system overview and the background technologies used, and then focusing on bilingual term alignment as the main contribution of this paper.

3.1 System overview

The TermEnsembler system extracts bilingual terminology from English and Slovenian texts, and stores it into a concept-based terminology database, meaning that the entries are organized to correspond to a concept (cf. the general theory of terminology proposed by Wüster (1979)), but a concept might have more than one corresponding designator. It is a semi-automated system, meaning that the user can select several extraction parameters and manually curate the monolingual

extraction results for better bilingual alignment. While the system currently supports two languages (English and Slovenian), additional languages can be added by implementing appropriate language-specific background technologies similar to the ones described in this paper. In addition to the extraction of individual terms in each of the two languages (extracted using the approach described in Section 3.2), it also stores aligned term pairs (aligned using the approach described in Section 3.3). We have also developed a method for extracting good examples and domains, but as these are additional functionalities, we refer the reader to the previous papers by Repar and Pollak (2017a, 2017b).

The system relies on several background resources and technologies, used in different components of the system:

- *Preprocessing:* Texts are extracted from the translation memory (TMX) and preprocessed using the part-of-speech tagger and Wordnet lemmatizer from NLTK (Bird et al. 2009) for English and using the ReLDI tagger and lemmatizer (Ljubešić and Erjavec 2016) for Slovenian.
- *Monolingual term extraction:* Monolingual term extraction method LUIZ-CF by Pollak et al. (2012), extending LUIZ (Vintar 2010), is used as a basis for our upgraded LUIZ-CF++ term extraction approach.
- *Bilingual term alignment:* We use the Pialign phrase table extraction functionality (Neubig et al. 2011) as a basis for implementing three different bilingual term alignment approaches PTBA-1, PTBA-2 and PTBA-3 used in our experiments. In the reimplementation of bilingual LUIZ, we use Giza++ for word alignment (Och and Ney 2003). For weight assignment in our ensemble approach, we use the evolutionary computation framework DEAP (Distributed Evolutionary Algorithms in Python) by Fortin et al. (2012).

The overall structure of the system is shown in Figure 1. The starting point is a bilingual corpus in the standard translation memory format TMX, from which also available metadata, such as term domain or language variety can be extracted. The text is extracted and preprocessed resulting in a list of aligned lemmatized and POS-tagged sentence pairs. These pairs are sent into the additional metadata extraction (e.g., when domain information is not available in the TMX) and the monolingual extraction process, which results in two separate monolingual term lists (for TL1 and TL2). At this point, these two term lists can be curated by the user of the system. The (raw or curated) term lists are then taken as input to the bilingual alignment process (described in detail in Figure 2), which produces the final list of aligned term pairs. Finally, these term pairs are entered in the termbase alongside the metadata extracted in the step described above.

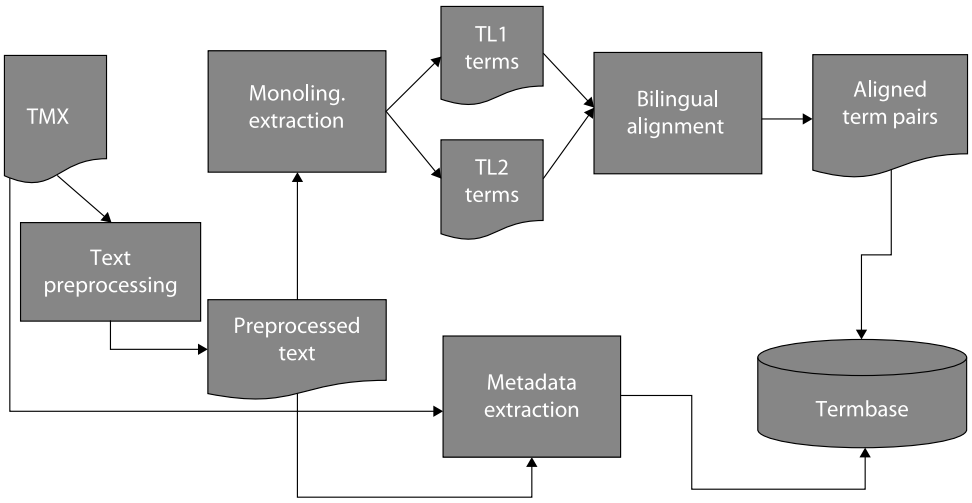

Figure 1. TermEnsembler: Methodology and components of the TermEnsembler system. Note that at several points human curation is possible (after monolingual extraction, after bilingual alignment or when accepting terms and metadata in the termbase. The monolingual step can also be skipped if the monolingual term lists are manually provided

3.2 Monolingual term extraction: LUIZ-CF++ upgrade of LUIZ-CF

The implemented monolingual term extraction approach LUIZ CF++ is based on the LUIZ hybrid approach by Vintar (2010) and refined with scoring and ranking functions implemented in LUIZ-CF by Pollak et al. (2012). The LUIZ approach is based on a list of part-of-speech patterns and a formula for comparison of term frequency between a domain corpus and a general language corpus (we used frequency lists from corpus Kres (Logar et al. 2012) for Slovenian and the British National Corpus (2007) for English).

In LUIZ-CF++, used in our experiments, we upgraded the LUIZ-CF monolingual term extraction approach by implementing the following additional functionalities:

- *Near-duplicates detection*: When importing the terms, the near duplicates (e.g. the orthography with or without spaces or hyphens, British and American English spellings) are detected and not created as new entries, but can be added as term variants of existing entries.
- *Nested term filtering*: According to Frantzi et al. (2000), nested terms are the terms that appear within other longer terms, and may or may not appear by themselves in the corpus. If the difference between a term and its nested term is below a certain threshold (which, in our case, can be defined by the user), only the longer term is returned. If not, both terms are included in the final output.

3.3 Bilingual term alignment: A novel ensemble learning approach

In this section, we describe the core part of TermEnsembler, i.e. the bilingual term alignment methodology implementing the *extract-align* approach explained in Section 2.2. Having implemented seven elementary term alignment approaches (3 existing, one modified, and 3 novel variants based on Pialign), this section introduces a novel ensemble-based approach combining the selected elementary term alignment approaches using an evolutionary algorithm.

We start by a brief outline of the proposed term alignment approach, illustrated in Figure 2. The input to the proposed TermEnsembler's bilingual term alignment methodology are two term lists (TL1 and TL2), which are automatically extracted using the monolingual extraction component (described in Section 3.2) or are human-defined. These two term lists are fed into seven individual bilingual term alignment algorithms that produce a total of 7 separate lists of aligned term pairs (*aligned term lists* or ATL), ranked by their alignment probability score as described in Section 3.3.1. The outputs of each alignment method are first normalized (separately) to the [0,1] interval, then fed into the evolutionary weights optimization algorithm described in Section 3.3.3 (which uses an external *ground truth list* (GTL) of manually annotated term pairs) to produce an optimal set of weights. These weights are then used to merge the seven ATLs into the final merged ATL using the procedure from Section 3.3.2.

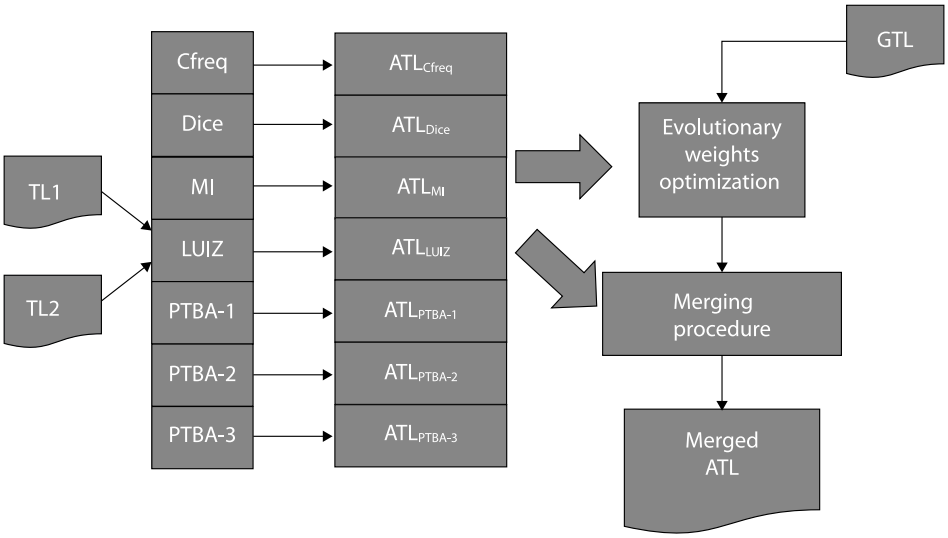

Figure 2. TermEnsembler's bilingual term alignment methodology

3.3.1 *Individual bilingual term alignment algorithms*

Each term alignment component described in this section produces a list of aligned term pairs ranked by their alignment scores, which are normalized between 0 and 1. The calculation of the scores is described below. The first four reimplemented approaches produce each one output (one aligned term list), while the last, novel approach, has three variants, leading to a total of seven output lists of aligned term pairs.

Co-frequency

Co-frequency *cofreq*(t_S, t_T) simply counts the number of sentences in which a term (t_S) from a source language S and a term (t_T) from target language T co-occur in the same sentence pair. The higher the co-occurrence count, the higher the probability that the terms are a correct term pair. This is the simplest of the used approaches and is completely language independent, but it does not take into account any language specifics. Because of that, it also requires a larger input corpus to produce sensible results.

Dice

This approach to bilingual terminology extraction is based on the Dice algorithm (Dice 1945). The co-frequency score from the previous component is used in the calculation of the Dice score, defined as follows:

$$dice(t_S, t_T) = 2 \frac{cofreq(t_S, t_T)}{freq(t_S) + freq(t_T)}$$

where (t_S) and (t_T) are source and target terms, respectively. The *freq(t)* function stands for the frequency of term t in the entire corpus. A score based on Dice is used also in Sketch Engine (Baisa et al. 2015).

Mutual information

Similar as Dice, MI (Church and Hanks 1990) calculates term alignment by taking into account the co-frequency of source and target terms and the individual frequency of each term. It is defined as follows:

$$MI(t_S, t_T) = \log_2 \frac{cofreq(t_S, t_T)}{freq(t_S)\, freq(t_T)}$$

It usually contains the multiplication with N (in our case the number of candidate terms), but since in our case N is constant across terms, we can omit it if we just want to rank the terms.

BI-LUIZ+

We used a modified version of the bilingual component of the LUIZ approach, described by Vintar (2010). This approach takes as input two lists of term candidates (one for the source language and one for the target language) and word alignment pairs (with probabilities). The original paper uses the Twente aligner (Hiemstra 1998), while we used the GIZA++ (Och and Ney 2003).[3]

Using the alignments, the best matches (1 or more) are computed for each source term as follows: given a source term, we iterate through all target terms. For each target term we compute a score by summing the probabilities that a target token is a translation of a source token. Note that in the original paper by Vintar (2010) the equivalence score takes all single-word probabilities and divides them by the number of words, but dividing is not performed in our re-implementation as in the testing phase it produced worse results.[4] If the score is non-zero, we add the target term to the list of candidates.

Novel Phrase-Table-Based Alignment (PTBA) approaches PTBA-1, PTBA-2 and PTBA-3

The proposed PTBA approaches are novel bilingual term alignment approaches that we have developed based on Pialign (Neubig et al. 2011), an unsupervised model for joint phrase alignment and extraction using nonparametric Bayesian methods and inversion transduction grammars. Pialign follows a similar approach to phrase table generation in statistical machine translation (SMT) (Koehn et al. 2007), however, instead of first generating word alignments and then extracting a phrase table consistent with these alignments, it joins the phases of alignment and extraction by constructing a generative model that includes phrases at many levels of granularity, from single words to full sentences. Similar to Haque et al. (2014), the PTBA approach uses machine translation phrase tables for term alignment, but differs from it in several aspects described below.

The proposed PTBA approach takes as input a corpus and produces the list of aligned terms as output. Specifically, the Pialign alignments are read and used for mapping that stores for each English word all the computed Slovenian alignments along with the frequency of each alignment. As illustration, take the following example:

manager → upravitelj (20%), upravljavec (30%), upravljavec premoženja (50%)

The same mapping is also created for the reverse direction (Slovenian to English). For each aligned sentence pair found to contain some English and Slovenian terms, we compute the matching of all English terms from this sentence against

3. We had to use a different alignment method since the Twente aligner does not work anymore.

4. In communication with Vintar it has been confirmed that division has been later excluded from the formula.

all phrases from this sentence, and the best matching is retained. The matching is computed as the ratio of the most similar substring (i.e. if the phrase contains the entire term, the result is 100%). As a result, for each English phrase found in a sentence we record which terms found in this sentence are a part of this phrase. The matching procedure is repeated also for Slovenian. Finally, for each sentence we retain only the term-to-phrase mappings that exist in both directions. That is, we store a mapping if an English term from some sentence matches an English phrase from the same sentence and a Slovenian term from the aligned Slovenian sentence matches with the aligned Slovenian phrase.

As a side result of this term-to-phrase matching procedure, we propose the following procedure to obtain a list of direct candidates for aligned terms (i.e. we identify the phrase alignments consisting of a single term). The conditions are that the best term-to-phrase matching score is at least 95% for English and 90% (as the language is morphologically more varied) for Slovene and the difference in length of term string and phrase string is not greater than 4. An example, where a term matches the phrase with nearly no differences is a term *upravitelj* and the phrase *upravitelji*. As this is the only element of the phrase, we assume that the aligned phrase is the term's equivalent in English (e.g. *manager*).

The matching problem is addressed as follows: For each sentence, we have a list of phrases in English, their aligned counterparts in Slovenian, a list of terms for each English phrase and a list of terms for each Slovenian phrase. When computing the matching between English and Slovenian terms we also take into account the possibility that the terms can consist of several words.

We define the matching score of a multi-word English term to a multi-word Slovenian term as the sum of best single word alignment scores among all word combinations between the terms. Consider the following example:

English sentence	*The name of the share class Allianz….*
Slovenian sentence	*Ime razreda delnic Allianz…*
English phrase	*The name of the share class*
Slovenian phrase	*Ime razreda delnic*
English terms	*share, share class*
Slovenian terms	*delnica, razred delnic*

The matching algorithm computes the sum of all best word alignment scores. For example *score(share, delnica) + score(class, razred)* is the alignment score for terms *share class* and *razred delnic* (the word (mis)alignments *share-razred* and *class-delnica* have very low or possibly zero scores and are not added to the sum).

The matching scores are accumulated for all phrases and all sentences. In the end, we obtain the probability distributions for the translation of English terms into Slovenian and Slovenian terms into English. Using this information, we can produce three translation tables: *symmetric, English to Slovenian, and Slovenian to English*, respectively. The symmetric table consists of only those aligned terms

where the greedy probabilistic translation is the same in both directions. That is, a pair of English and Slovenian terms have each other listed as the most probable translation. The other two translation tables simply list the most likely translation in each direction. In this way, we have defined three different PTBA term alignment methods, resulting in three separate outputs of the PTBA term alignment method:

- *PTBA-1 Aligned Term list*, containing the results of the symmetric translation table.
- *PTBA-2 Aligned Term list,* containing the results of the English to Slovenian and Slovenian to English translation tables.
- *PTBA-3 Aligned Term list,* containing the list of direct alignment candidates produced as a side result of the term-to-phrase matching procedure.

3.3.2 *Final term pair ranking by ensemble-based weighting of separate lists of term pairs*

This section presents the key part of the developed methodology for ranking of aligned term pairs, i.e. the mechanism for assigning weights to separate lists of term pairs obtained by individual term alignment algorithms, and the merging mechanism using an ensemble weighting approach.

The ensemble score (Escore) is computed from two separate weighting scores:

- the algorithm weight (w), and
- the term pair score (score), normalized to [0,1].

A merging procedure for computing the final ensemble score Escore takes the individual term pair scores (score) from each of the seven elementary algorithms, together with weights for each approach provided by the user or assigned by automated means (i.e. the evolutionary algorithm approach explained below) and returns the final aligned term list, re-normalized on the [0,1] interval.

Merging procedure

1. For all **term pairs** (t_S, t_T) **compute** $Escore(t_S, t_T)$:

$$\begin{aligned} Escore(t_S, t_T) = {} & w_{cofreq}\, score_{cofreq}(t_S, t_T) + \\ & w_{dice} \cdot score_{dice}(t_S, t_T) + \\ & w_{mi} \cdot score_{mi}(t_S, t_T) + \\ & w_{mi} \cdot score_{mi}(t_S, t_T) + \\ & w_{luiz} \cdot score_{luiz}(t_S, t_T) + \\ & w_{PBA_1} \cdot score_{PTBA\text{-}1}(t_S, t_T) + \\ & w_{PBA_2} \cdot score_{PTBA\text{-}2}(t_S, t_T) + \\ & w_{PBA_3} \cdot score_{PTBA\text{-}3}(t_S, t_T) \end{aligned}$$

2. Compute *Normalized Escore*(t_S, t_T)∈*[0,1]*
3. Rank *term pairs* (t_S, t_T) in decreasing order of their *Normalized Escore*(t_S, t_T)

3.3.3 *Evolutionary weighting of term alignment algorithms*

To be able to effectively search the large space of various weight values, we decided to use an evolutionary algorithm to find an optimal configuration. Specifically, we utilized the genetic algorithm (GA) implementation in DEAP (*Distributed Evolutionary Algorithms in Python*) by Fortin et al. (2012), an evolutionary computation framework, which can be used for rapid prototyping and testing of ideas and is designed to make algorithms explicit and data structures transparent. The GA algorithm starts with a random population and then applies crossover (producing new (children) members of the population from existing (parent) members) and mutation (randomly changing individual members – similar to biological mutation) operations for a successive number of generations. In each generation, the children are evaluated using a custom evaluation function and those that perform better than the parents are retained, while those that perform worse are discarded which eventually leads to an optimal result.

We start by generating a population of random sets of seven real numbers in the form of 7-tuples of weights of the 7 individual bilingual term alignment outputs:

$$(w_{cofreq}, w_{dice}, w_{mi}, w_{luiz}, w_{PBA1}, w_{PBA2}, w_{PBA3})$$

Each 7-tuple is used to generate a final bilingual term list (see Section 3.3.2) and is evaluated against a database of manually annotated term pairs provided in the training dataset. We used the parameters suggested in the DEAP documentation: number of generations: 100; population: 100; crossover probability: 0.5; mutation probability: 0.2.

We repeated the GA algorithm execution 20 times, and then calculated the average precision and standard deviation of the best performing 7-tuple of weights in each GA repetition. We selected the overall best performing 7-tuple learned on the training domain (training dataset) and tested its performance on two separate domains (test datasets). DEAP can be set up to optimize a single objective (i.e. precision among the Top 400 term pairs as in Section 4.4.1) or multiple objectives (i.e. precision among the Top 400 term pairs and number of correct *multi-word unit* (MWU) term pairs as in Section 4.4.2) at the same time.

4. Experiments and results

This section describes the experiments conducted to evaluate the TermEnsembler bilingual term alignment methodology and the datasets used in the experiments, followed by the results of the experiments and a qualitative analysis of errors.

4.1 Experimental setting

In these experiments, our goal was to find the best weight configuration for the 7 outputs produced by the individual term alignment components. To do so, we first evaluated the outputs individually in terms of overall precision and precision of MWU (*multi-word unit*) terms and then tried to find the best weight configuration using the evolutionary algorithm. We learned the best weight configuration on one domain (*Financial*) and then tested it on two others, non-related domains (*IT* and *Automotive*), by which we show that it is applicable to different domains.

The experimental setting was as follows. In creating the monolingual term lists as described in Section 3.2, we included only the terms that appear more than 10 times in the dataset.

– The evaluation criterion was the precision of term alignment, where the criterion for annotation was proper alignment, and not whether the individual English and Slovenian units are actually terms or not.

The latter requires further clarification.

– As bilingual term alignment is the main focus of this paper, we were primarily concerned with whether the terms are aligned properly (whether the terms are translation equivalents) and not whether the terms are true terms in each language.[5] For illustration, consider the following two examples:

  exchange rate – menjalni tečaj
  end of march – konec marca

  In the first example, both terms (English and Slovenian) are true terms according to the definition of a term from ISO 1087 ("verbal designation of a general concept in a specific subject field"), while the terms in the second example are much less likely to be considered terms in the sense of ISO 1087. However, for the purposes of evaluating the bilingual alignment algorithm both examples were considered correct.

5. An evaluation by a subject-matter expert reviewing the top 200 term pairs produced by the system showed that 74.5% of them are true terms.

– The evaluation was performed by a single annotator, which is the only realistic setting in a language-industry environment. Nevertheless, for inter-annotator evaluation, we acquired a second annotator to annotate a subset of the final output produced (and previously annotated by the main annotator) with the final weight configuration (see Section 4.4) on the Financial domain. The inter-annotator agreement was high, with both annotators agreeing in more than 95% of term pairs and Cohen's kappa (Cohen 1968) reaching 0.900. This denotes almost perfect agreement according to Landis and Koch (1977), and we can safely assume that annotations performed by a single annotator are highly accurate.

Note that in addition to measuring the precision of term alignment, we initially also considered measuring the recall, for which we would need a dataset containing manually annotated term pairs. However, measuring recall proved to be practically less relevant. The client arrived at the conclusion that in a production environment of a language service provider, the recall is not of particular importance, while it is much more important that term extraction output be precise, requiring no or minimal further processing or manual selection. As will be shown in Section 4.4, TermEnsembler produces a large number of correct term pairs, which satisfies the needs of the client. However, for the purpose of this article, we did evaluate the recall on a small gold standard term list in Section 4.4.3.

4.2 Data

In our experiments we used three distinct datasets, all coming from a production environment of a language service provider.

– Financial. This translation memory contains segments from a long-term translation project in the financial domain, specifically annual reports of investment funds and various related documentation. It has 18,197 segments (i.e. bilingual segment pairs) with 396,295 words in English and 354,862 words in Slovenian. The default configuration of the monolingual extractor returned 1,723 English and 1,953 Slovenian terms. This dataset was used to find the best weight configuration with the evolutionary algorithm.
– IT. This translation memory was used in a long-term software localization project. Most segments contain user interface strings and a smaller portion also contains user assistance (i.e. help articles) content. It has 40,599 segments (i.e. bilingual segment pairs) with 523,819 words in English and 473,430 words in Slovenian. The default configuration of the monolingual extractor returned 2,234 English and 2,477 Slovenian terms. This dataset was used to test the best weight configuration found with the evolutionary algorithm on the Financial dataset.

– Automotive. This translation memory was used in a long-term project for a customer from the automotive industry and contains segments from user manuals, internal service documentation and customer-facing promotional materials. It has 65,516 segments (i.e. bilingual segment pairs) with 861,665 words in English and 779,145 words in Slovenian. The default configuration of the monolingual extractor returned 3,122 English and 3,879 Slovenian terms. This dataset was used to test the best weight configuration found with the evolutionary algorithm on the Financial dataset.

Detailed statistics for each dataset, including the number of terms obtained by monolingual terminology extraction, are presented in Table 1.

Table 1. Detailed statistics of the three datasets used in the experiments

	Financial	IT	Automotive
Total segments	18,197	40,599	65,516
Total English words	396,295	523,819	861,665
Total Slovenian words	354,862	473,430	779,145
Unique English words	11,365	21,711	25,591
Unique Slovenian words	20,093	31,973	43,406
English terms	1,723	2,234	3,122
Slovenian terms	1,953	2,477	3,879

4.3 Experimental comparison of individual bilingual term alignment components

In this section, we systematically compare the performance of individual bilingual term alignment components from two aspects. First, we focus on the overall precision of the Top N term pairs produced by each component, and then we turn our attention to MWU (*multi-word unit*) term pairs found in the top N term pairs produced by the individual components.

4.3.1 *Precision of individual term alignment components*

Table 2 provides the results for precision for each method on the Financial dataset. We can observe that two PTBA methods have the highest precision, followed by another PTBA method and the three frequency-based components (Co-frequency, Dice and Mutual information), while BI-LUIZ+ has the lowest precision.

Table 2. Precision of individual bilingual alignment components on the Financial dataset on the Top 100, Top 200, Top 400 and Top 800 term pairs according to their (normalized) alignment score

	Total term pairs	Top 100		Top 200		Top 400		Top 800/ Total	
		Corr.	Prec.	Corr.	Prec.	Corr.	Prec.	Corr.	Prec.
Co-freq	1,492	60	0.600	111	0.555	175	0.438	292	0.366
Dice	1,492	57	0.570	128	0.640	272	0.680	511	0.693
MI	1,492	59	0.590	120	0.600	229	0.573	398	0.498
BI-LUIZ+	1,561	43	0.430	82	0.410	136	0.340	228	0.285
PTBA-1	591	93	0.930	183	0.915	350	0.875	**486**	**0.822**
PTBA-2	1,341	74	0.740	148	0.740	246	0.616	436	0.546
PTBA-3	674	**98**	**0.980**	**193**	**0.965**	**360**	**0.900**	523	0.777

Note that since the total number of term pairs of PTBA-3 and PTBA-1 is lower than 800, the last column denotes precision on the total number of pairs (i.e. Top 674 and Top 591, respectively).

4.3.2 *Single vs. multi-word unit terms*

While precision is the most important performance indicator of a bilingual term alignment algorithm, we also wanted to have more details on the ratio between single and multi-word terms in the outputs, because the client communicated that having translations of multi-words terms is much more useful than just simple one-word units. Since we are looking at bilingual term pairs, we consider a pair to be a single-word unit if both terms (English and Slovenian) are single-word units, and multi-word if at least one of the terms is a multi-word unit (MWU). For illustration, see the three examples below:

issuance – izdaja *SINGLE-WORD UNIT*
registrar – agent za registracijo *MULTI-WORD UNIT*
stock market – borzni trg *MULTI-WORD UNIT*

Specifically, we looked at how many of the top N terms produced by individual components are correct MWU term pairs. This decision was again reached in communication with the client who wanted to have the ability to request a specific number (N) of term pairs to be returned by TermEnsembler and our goal was to make the returned term pairs as good as possible, both in terms of overall precision and in the number of correct MWU terms.

In Table 3, we can observe that the Dice algorithm produces the most correct term pairs in all 4 scenarios, closely followed by MI. BI-LUIZ+ produces a lot of multi-word terms but its precision (calculated as correct MWU terms divided by all MWU terms in the top N term pairs) is relatively low, while the PTBA methods

produce few MWU term pairs in the Top 100 pairs, but improve in this respect in Top 200, Top 400 and Top 800 scenarios.

Table 3. Total number of MWU term pairs (and their precision) in top N terms, correct MWU term pairs on the Financial dataset

	Top 100		Top 200		Top 400		Top 800/Total	
	Cor/tot	**Prec**	**Cor/tot**	**Prec**	**Cor/tot**	**Prec**	**Cor/tot**	**Prec**
Co-freq	2/21	0.420	7/49	0.143	17/128	0.133	49/383	0.128
Dice	**52**/94	0.553	**106**/175	0.606	**198**/320	0.619	**358**/589	0.608
MI	50/87	0.575	102/178	0.573	187/351	0.533	295/678	0.435
BI-LUIZ+	43/100	0.430	82/200	0.410	103/363	0.284	136/680	0.200
PTBA-1	20/24	0.833	51/61	0.836	133/170	0.782	199/273	0.729
PTBA-2	15/38	0.395	39/85	0.459	90/234	0.385	194/527	0.368
PTBA-3	14/14	**1.000**	54/57	**0.947**	130/146	**0.890**	218/278	**0.784**

Note that since the total number of term pairs of PTBA-3 and PTBA-1 is lower than 800, the last column denotes precision on the total number of pairs (i.e. Top 674 and Top 591, respectively).

4.4 Results of the TermEnsembler's bilingual term alignment approach

The key question in our system is how to determine the optimal configuration of weights for the merging script described in Section 3.3. Table 2 and Table 3 above clearly show that some of the methods are much more effective than the others. Similar to the reasoning in Section 4.3, we want to test two distinct scenarios:

- In the first one, we want to find the best overall precision.
- In the second one, we want to find the best compromise between the overall precision and the number of correct multi-word units.

We decided to focus the evaluation of the weight configuration on the top 400 term pairs, because the client believes that 400 terms are enough to produce a useful terminological resource in a standard translation project. In other words, we try to optimize the configuration to return the best results on the top 400 term pairs. Also, the starting point for comparison is the result of the PTBA-3 component that has an overall precision of 0.900 and returns 130 correct multi-word unit term pairs (see Table 2). This means that any weight configuration would need to improve on these results.

As evident from Table 4, assigning the same weight to all components does not yield results superior to the PTBA-3 component. The same is true if we assign weights according to their individual precision (calculated in Table 2) relative to the lowest value (i.e. the weight of BI-LUIZ+ is 1.0 and the rest are calculated

proportionally). This is why we decided to use the DEAP evolutionary algorithm described in Section 3.3 for weight configuration.

4.4.1 *Optimizing for optimal precision*

In the first experiment, we wanted to construct a weight configuration that would result in the highest possible precision, which means that we minimize the number of incorrect pairs. We performed 20 repetitions of the evolutionary algorithm execution. The average precision of the best performing 7-tuples of weights in each of the 20 repetitions was 0.949 with a standard deviation of 0.009. The overall best precision of 0.960 was achieved by three different weight configurations (see Table 5),[6] showing that the evolutionary algorithm exceeds the results of PTBA-3 by 6% (see Table 4).

Table 4. Results of the various weight configurations on the Financial domain

	Top 400
PTBA-3	0.900
Equal weights	0.725
Precision weights	0.732
Evolutionary algorithm	**0.960**

To test whether this configuration can be applied universally, we used it to evaluate precision on two additional domains: *Automotive* and *IT*. To do so, we tested all three configurations from Table 5 and calculated the average overall precision. As can be observed from Table 6, the weight configuration produced by the evo-

6. The calculated weights show that the PTBA-3 component is always the most significant one, followed by PTBA-1, and next Cofreq followed by all other methods (which can in some cases even have negative weights). Several factors that contribute to the actual magnitude of weights have to be taken into account when interpreting the results. First, the weights are computed using different heuristics. Second, the components produce results of different lengths and those returning a small number of mostly correct results are likely to obtain a higher weight. Next, the evolutionary algorithm will try to adjust the weights in such way that segments of high ranked correct results will make it to the final list. If the same or similar segment of correct results appears at the bottom of the list of another component, its promotion to the final list is likely to be too costly as this would also promote several incorrect results. For example, the reason for the negative weights in some of the repetitions in Table 5 is that the scores assigned by a particular component (i.e. PTBA-2) are too high compared to other components. This is confirmed by the results of the manual evaluation of individual components in Table 2 where we can observe that PTBA-2 has a significantly lower precision than PTBA-1 or PTBA-3. The weights of the remaining 4 components are significantly lower, close to 0, with the highest one of them being Cofreq.

Table 5. The best performing weight configurations when optimizing overall precision using an evolutionary algorithm

Rep #	Cofreq	Dice	MI	Luiz	PTBA-1	PTBA-2	PTBA-3
3	0.619	0.196	0.010	0.053	4.481	−2.867	11.046
8	0.327	0.086	0.008	0.022	1.564	0.137	5.494
10	0.561	0.106	−0.017	0.104	2.177	−0.758	10.268

lutionary algorithm returns good results on unseen data (*IT* and *Automotive*) as well, with precision on unseen data actually exceeding the precision on the training data (i.e. *Financial* domain).

Table 6. Precision of the weight configuration produced by the evolutionary algorithm on the Financial domain and applied to the Automotive and IT domain. The results were obtained as an average precision of the three weight configurations shown in Table 5

	Top 400
Financial	0.960±0.000
Automotive	0.984±0.001
IT	0.984±0.001

4.4.2 *Optimizing for a compromise between optimal precision and number of correct multi-word unit term pairs*

In the next step, we modified the evolutionary algorithm to optimize the configuration for the highest precision and the largest number of multi-word units at the same time. While the equal weight configuration and the weight configuration based on individual precision values produce a higher number of MWUs, they also introduce a fair amount of noise resulting in lower precision. As is evident from Table 7, the configuration produced by the evolutionary algorithm has the highest precision while maintaining a decent amount of MWUs (a high number of which are also correct – MWU precision of 0.919). The results closest to this configuration are returned by the PTBA-3 component, but the number of MWUs is significantly lower.

These results were achieved by running 20 repetitions of the evolutionary algorithm and selecting the best weight configuration based on the following criterion: the best configuration has the highest number of correct MWUs and must have an overall precision greater than the best individual component (in our case, PTBA-3). The best weight configuration was thus produced in repetition 19 and had the weights shown in Table 8.

Table 7. Overall precision, total number of MWUs, number of correct MWUs and precision of MWUs of the configuration produced by the evolutionary algorithm compared to various other configurations, measured on the Financial domain

	Precision	Total MWUs	Correct MWUs	MWU precision
PTBA-3	0.900	146	130	**0.890**
Equal weights	0.725	311	205	0.659
Precision weights	0.733	**312**	**208**	0.667
Evolutionary algorithm	**0.955**	185	170	0.919

Table 8. The best performing weight configuration when optimizing for a compromise between optimal precision and number of correct multi-word unit term pairs

Rep #	Cofreq	Dice	MI	Luiz	PTBA-1	PTBA-2	PTBA-3
19	0.219	0.229	0.009	0.116	2.855	−4.739	11.470

Once again, we tested whether the configuration produced by the evolutionary algorithm can be used universally by applying it to two additional domains: Automotive and IT. The results can be found in Table 9.

Table 9. Top 400 results of the weight configuration produced by the evolutionary algorithm on the Financial domain and applied to the Automotive and IT domain

	Precision	Total MWUs	Correct MWUs	MWU precision
Financial	0.955	**185**	**170**	0.919
Automotive	**0.990**	153	151	**0.987**
IT	0.985	130	126	0.969

In both domains, the results are similar to what we observed in the *Financial* domain. In fact, the results are even better in the two new domains with overall precision in the Top 400 term pair candidates exceeding 98%, and the MWU precision above 96%. The actual ratio of correct MWU terms among the Top 400 terms is 38% on the *Automotive* domain and 32% on the *IT* domain. We decided to use this configuration as the final configuration in the client's production environment.

4.4.3 *Recall of the TermEnsembler system*

Due to the client's preference, the majority of our experiments were focused on precision, but we did also evaluate recall on a corpus subsample, where a gold standard termlist of 88 financial terms was produced by manual expert annotation. With the final weight configuration (used in the production environment) the recall of the TermEnsembler system was 60%.

4.5 Qualitative analysis of errors

To better understand the types of errors that the system makes, for each of the three domains we have performed a qualitative analysis of the first 50 incorrect term pairs[7] among the list of 800 top ranked term pairs suggested by the system, using the final weight configuration suggested by the evolutionary algorithm. We observed that most of the errors are due to discrepancies between the English and Slovenian monolingual extraction process, rather than due to the incorrect alignment procedure, and that many incorrect term pairs can be considered "partially correct". We illustrate several examples of incorrect alignments below, starting with minor errors followed by some more severe cases of misaligned terms.

In some of the highly ranked term pairs, one part of a term in one language is missing because the term was incorrectly extracted, which results in partially correct term pair, such as (the word in brackets was not extracted):

Financial: *interest (rate) – obrestna mera*
Automotive: *(quick) repair kit – komplet za hitro popravilo*
IT: *missing (value) – manjkajoča vrednost*

A particularly difficult issue for the system are product names. Because they may not follow standard language rules regarding the construction of terms, they are difficult to detect without a pre-defined product name list or a well performing named entity recognition system. Consequently, many of the incorrectly extracted named entities contain parts of product names. The Financial dataset in particular has a high number of named entities, which is a reason for lower results compared to the other two corpora. Such examples include:

Equity – delnica[8]
BNP Paribas – Paribas
Flexible Bond Strategy – Bond Strategy

In a limited number of cases, the monolingual terms and the alignment itself are correct, but the resulting term pair is not correct. In the two examples from the Automotive dataset, the source text uses *miles per gallon* to denote gas mileage, but the Slovenian translation (due to the preferences of the customer) uses *kilometers per 100 liters*. A similar case can be observed with units denoting weight.

Mile – km
Lb – kg

7. The positions of the 50th incorrect term pair for all three domains: 518 for Financial, 756 for Automotive, and 661 for IT.

8. Note that "equity" can appear either as a common noun (i.e. equity = assets) or as a part of a proper noun (e.g., Global Equity Climate Change).

In a smaller number of cases close to the bottom of the list of extracted term pairs, the alignment is completely off and the meaning of the source term is not the same as the meaning of the target term (which can be explained by the frequent co-occurrence of the terms in the text), for example:

Financial: *gross national income – svetovna banka*
Automotive: *similar heavy object – pritrjen nosilec koles*
IT: *folder number – znesek kredita*

Finally, we compared the ratio between the two major error types in the three domains (see Table 10). In the *Financial* and *Automotive* domains, the majority of the incorrect terms can be ascribed to the category "Partially correct", which are predominantly errors arising from incorrect monolingual extraction (but could also be related to incorrect translation or wrong alignment of the two terms). Because the monolingual term is missing a word or several words or contains redundant words, the resulting term pair was not classified as correct. However, the alignment is not completely wrong nor completely useless, because the term can be quickly corrected in a semi-automated terminology setting.

Table 10. A comparison of the two major error type among the 50 analysed incorrect term pairs

	Financial	**Automotive**	**IT**
Different meaning	38%	12%	**56%**
Partially correct	**62%**	**88%**	44%

5. Conclusions and future work

This paper describes TermEnsembler, a terminology extraction and alignment system, created from the point of view of language service providers in the language and translation industry. It consists of a concept-oriented terminology database with industry-standard file format support for easy sharing with other terminological applications, an online user interface for database management and semi-automatic term extraction, a monolingual terminology extraction algorithm (currently supporting English and Slovenian) and a novel bilingual alignment methodology with several components.

The first step is monolingual extraction based on the work of Vintar (2010) and Pollak et al. (2012) with some additional modifications, such as a filter for nested terms and near-duplicate recognition. The final result of this step are two lists of terms (one for each language) with the terms ordered by their termhood score. The next step, which is the central part of the paper, involves bilingual

alignment of the terms in the two lists. We have implemented and evaluated a total of seven methods – implementing approaches from the related work and the newly proposed approaches – which all return a list of aligned English-Slovenian term pairs. The evaluation of each approach separately shows that the highest precision was obtained by the newly developed phrase-table-based term alignment approach PTBA-3 which directly matches the extracted terms with phrases from the phrase table.

For final implementation, we experimented with different merging methods for the 7 outputs by assigning weights to produce a final list of term pairs. After initial experiments with equal weight and precision-based weights, we opted for an ensemble optimization approach using the genetic algorithm implementation from the evolutionary algorithm framework DEAP by Fortin et al. (2012), which takes random weight configurations and tries to optimize them towards a certain goal over a successive number of generations.

We have trained the bilingual alignment approach in TermEnsembler on one domain and tested it on two different domains achieving excellent results, with more than 96% of the top 400 term pair alignments produced by the system evaluated as correct by a human evaluator. In addition, we have also tried to optimize the system for producing a greater number of multi-word terms because they are particularly complicated for translation. When optimizing the evolutionary algorithm for overall precision and number of correct multi-word terms, at least a third of the top 400 term pair alignments returned by our system were correct multi-word terms, with precision computed on the MWUs reaching 0.919. All in all, we believe the high precision of our system among the top 400 terms would require only minor manual human curation to produce a viable term list for day-to-day work in the language industry.

We also briefly looked into whether bilingual term alignment improves the quality of monolingual terms. An experienced translator compared the top 200 terms returned by the initial algorithm (the LUIZ-CF variant described in Pollak et al. (2012)) for each of the two languages in all three domains and compared them with the top 200 terms produced by TermEnsembler after bilingual term alignment. The results show that TermEnsembler does improve the monolingual quality of terms (precision) by around 10%.

In terms of future work, we have identified several lines of research. We will continue adding new languages, implementing and systematically evaluating different monolingual term-extraction approaches. For bilingual alignment, we will initially focus on a systematic optimization of the evolutionary algorithm parameters and then look into implementing user-friendly parameters that would allow the users to tweak the weights towards greater overall precision or larger number of MWU terms. We will also test other, potentially faster optimization methods such

as differential evolution and Newton-like methods as well as develop machine-learning solutions for term alignment, combining the proposed statistical scores and cognate-based features, as in Aker et al. (2013). Finally, given a recent trend of well performing word-embeddings methods leading to excellent results in various natural-language processing tasks, we aim to address bilingual term-extraction as a well-suited task for developing cross-lingual embedding based term alignment methods, stimulated by the work of Conneau et al. (2018).

Acknowledgements

The system's interface and the elementary term extraction approaches were designed and developed in the scope of the TermIolar project by the Jožef Stefan Institute and Iolar d.o.o. The authors acknowledge the contribution of Simon Bratina and Davorin Sečnik (of Iolar d.o.o.) to functional specifications, additional requirements, evaluation of the interim results and providing important feedback and suggestions. The authors thank also Špela Vintar for her clarifications in the reimplementation of bilingual LUIZ term alignment.

The authors acknowledge the financial support of Slovenian Research agency for funding part of this research in the scope of basic research program Knowledge Technologies (Grant No. P2-0103) and the project Terminology and Knowledge Frames across Languages (Grant No. J6-9372). This paper is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No. 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this paper reflect only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

References

Ahmad, Khurshid, Lee Gillam, and Lena Tostevin. 2000. "Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER)." In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, 717–724. Washington, USA.

Aker, Ahmet, Monica Paramita, and Rob Gaizauskas. 2013. "Extracting Bilingual Terminologies from Comparable Corpora." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 402–411. Sofia, Bulgaria.

Amjadian, Ehsan, Diana Inkpen, Tahereh Paribakht, and Farahnaz Faez. 2016. "Local-Global Vectors to Improve Unigram Terminology Extraction." In *Proceedings of the 5th International Workshop on Computational Terminology*, 2–11. Osaka, Japan.

Baisa, Vít, Barbora Ulipová, and Michal Cukr. 2015. "Bilingual Terminology Extraction in Sketch Engine." In *9th Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2015 – Proceedings*, 61–67. Karlova Studánka, Czech Republic.

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Sebastopol: O'Reilly Media Inc.

Church, Kenneth Ward, and Patrick Hanks. 1990. "Word Association Norms, Mutual Information, and Lexicography." *Computational Linguistics* 16 (1): 22–29.

Cohen, Jacob. 1968. "Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit." *Psychological Bulletin* 70 (4): 213. https://doi.org/10.1037/h0026256

Conneau, Alexis, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. "Word Translation Without Parallel Data." (https://arxiv.org/abs/1710.04087) Accessed 2 February 2019.

Daille, Béatrice, and Emmanuel Morin. 2005. "French-English Terminology Extraction from Comparable Corpora." In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*, 707–718. Jeju Island, South Korea.

Daille, Béatrice, Éric Gaussier, and Jean-Marc Langé. 1994. "Towards Automatic Extraction of Monolingual and Bilingual Terminology." In *Proceedings of the 15th Conference on Computational linguistics*, 515–521. Kyoto, Japan. https://doi.org/10.3115/991886.991975

Dice, LR. 1945. "Measures of the Amount of Ecologic Association between Species." *Ecology* 26 (3): 297–302. https://doi.org/10.2307/1932409

Foo, Jody. 2012. *Computational Terminology: Exploring Bilingual and Monolingual Term Extraction*. Linköping: Linköping University Electronic Press.

Fortin, Félix-Antoine, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. "DEAP: Evolutionary Algorithms Made Easy." *Journal of Machine Learning Research* 13 (no. Jul): 2171–2175.

Frantzi, Katerina, Sophia Ananiadou, and Hideki Mirna. 2000. "Automatic Recognition of Multi-Word Terms:. the C-Value/NC-Value Method." *International Journal on Digital Libraries* 3(2): 115–130. https://doi.org/10.1007/s007999900023

Gouadec, Daniel. 2007. *Translation as a Profession*. Amsterdam/Philadephia: John Benjamins. https://doi.org/10.1075/btl.73

Haque, Rejwanul, Sergio Penkale, and Andy Way. 2014. "Bilingual Termbank Creation via Log-Likelihood Comparison and Phrase-Based Statistical Machine Translation." In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, 42–51. Dublin, Ireland. https://doi.org/10.3115/v1/W14-4806

Hazem, Amir, and Emmanuel Morin. 2017. "Bilingual Word Embeddings for Bilingual Terminology Extraction from Specialized Comparable Corpora." In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, 685–693. Taipei, Taiwan.

Hiemstra, Djoerd. 1998. "Multilingual Domain Modeling in Twenty-One: Automatic Creation of a Bi-Directional Translation Lexicon from a Parallel Corpus." In *Proceedings of the 8th CLIN Meeting*, 41–58. Amsterdam, The Netherlands.

Justeson, John, and Slava Katz. 1995. "Technical Terminology: some Linguistic Properties and an Algorithm for Identification in Text." *Natural Language Engineering* 1 (1): 9–27. https://doi.org/10.1017/S1351324900000048

Kageura, Kyo, and Bin Umino. 1996. "Methods of Automatic Term Recognition: A Review." *Terminology* 3 (2): 259–289. https://doi.org/10.1075/term.3.2.03kag

Khan, Muhammad Tahir, Yukun Ma, and Jung-jae Kim. 2016. "Term Ranker: A Graph-Based Re-Ranking Approach." In *Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference*, 310–315. Key Largo, USA.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan et al. 2007. "Moses: Open Source Toolkit for Statistical Machine Translation." In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 177–180. Prague, Czech Republic. https://doi.org/10.3115/1557769.1557821

Kupiec, Julian. 1993. "An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora." In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, 17–22. Columbus, USA. https://doi.org/10.3115/981574.981577

Landis, Richard, and Gary Koch. 1977. "The Measurement of Observer Agreement for Categorical Data." *Biometrics* 33 (1): 159–174. https://doi.org/10.2307/2529310

Ljubešić, Nikola, and Tomaž Erjavec. 2016. "Corpus vs. Lexicon Supervision in Morphosyntactic Tagging: the Case of Slovene." In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, 23–28. Portorož, Slovenia.

Logar, Nataša, Miha Grčar, Marko Brakus, Tomaž Erjavec, Špela Arhar Holdt, and Simon Krek. 2012. *Korpusi slovenskega jezika Gigafida, KRES, ccGigafida in ccKRES: gradnja, vsebina, uporaba [Slovenian language corpora Gigafida, KRES, ccGigafida, ccKRES: creation, content, use]*. Ljubljana: Trojina, zavod za uporabno slovenistiko; Fakulteta za družbene vede.

Macken, Lieve, Els Lefever, and Veronique Hoste. 2013. "Texsis: Bilingual Terminology Extraction from Parallel Corpora using Chunk-Based Alignment." *Terminology* 19 (1): 1–30. https://doi.org/10.1075/term.19.1.01mac

McEnery, Tony, Richard Xiao, and Yukio Tono. 2006. *Corpus-Based Language Studies: An Advanced Resource Book*. London: Taylor & Francis.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." (https://arxiv.org/abs/1301.3781) Accessed 10 July 2018.

Neubig, Graham, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. "An Unsupervised Model for Joint Phrase Alignment and Extraction." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 632–641. Portland, USA.

Och, Franz Josef, and Hermann Ney. 2003. "A Systematic Comparison of Various Statistical Alignment Models." *Computational Linguistics* 29 (1): 19–51. https://doi.org/10.1162/089120103321337421

Pollak, Senja, Anže Vavpetič, Janez Kranjc, Nada Lavrač, and Špela Vintar. 2012. "NLP Workflow for On-Line Definition Extraction from English and Slovene Text Corpora." In *Proceedings of KONVENS 2012*, 53–60. Vienna, Austria.

Repar, Andraž, and Senja Pollak. 2017a. "Good Examples for Terminology Databases in Translation." In *Electronic Lexicography in the 21st century. Proceedings of eLex 2017 Conference*, 651–661. Leiden, Netherlands.

Repar, Andraž, and Senja Pollak. 2017b. "Ontology-Based Translation Memory Maintenance." In *Proceedings of the 20th International Multiconference Information Society 2017*, 19–22. Ljubljana, Slovenia.

Schmitz, Klaus Dirk, and Daniela Straub. 2016. "Tight Budgets and a Growing Number of Languages Impede Terminology Work." *tcworld magazine for international information management* (http://www.tcworld.info/e-magazine/technical-communication/article/tight-budgets-and-a-growing-number-of-languages-impede-terminology-work/). Accessed 24 August 2018.

The British National Corpus, version 3 (BNC XML Edition). 2007. *Distributed by Bodleian Libraries*, University of Oxford, on behalf of the BNC Consortium. (URL: http://www.natcorp.ox.ac.uk/). Accessed 10 March 2017.

Vintar, Špela. 2010. "Bilingual Term Recognition Revisited. The Bag-of-Equivalents Term Alignment Approach." *Terminology* 16 (2): 141–158. https://doi.org/10.1075/term.16.2.01vin

Wang, Rui, Wei Liu, and Chris McDonald. 2016. "Featureless Domain-Specific Term Extraction with Minimal Labelled Data." In *Proceedings of the Australasian Language Technology Association Workshop*, 103–112. Melbourne, Australia.

Wermter, Joachim, and Udo Hahn. 2005. "Paradigmatic Modifiability Statistics for the Extraction of Complex Multi-Word Terms." In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 843–850. Vancouver, Canada.

Wüster, Eugene. 1979. *Introduction to the General Theory of Terminology and Terminological Lexicography*. Vienna: Springer.

Zhang, Zigi, Jie Gao, and Fabio Ciravegna. 2018. "SemRe-Rank: Incorporating Semantic Relatedness to Improve Automatic Term Extraction Using Personalized PageRank." (https://arxiv.org/abs/1711.03373) Accessed 7 January 2019.

Address for correspondence

Andraž Repar
International Postgraduate School Jožef Stefan
Jožef Stefan Institute
Jamova 39, Ljubljana
Slovenia
repar.andraz@gmail.com

Co-author information

Vid Podpečan
Jožef Stefan Institute
vid.podpecan@ijs.si

Anže Vavpetič
Jožef Stefan Institute
hi@anzevavpetic.com

Nada Lavrač
Jožef Stefan Institute
nada.lavrac@ijs.si

Senja Pollak
Jožef Stefan Institute
senja.pollak@ijs.si

Karst exploration: Extracting terms and definitions from karst domain corpus

Senja Pollak[1,2], Andraž Repar[1], Matej Martinc[1], Vid Podpečan[1]

[1]Jožef Stefan Institute, Ljubljana, Slovenia
[2]Usher Institute of Population Health Sciences and Informatics, Edinburgh Medical School, Edinburgh, UK
E-mail: senja.pollak@ijs.si, repar.andraz@gmail.com, matej.martinc@ijs.si, vid.podpecan@ijs.si

Abstract

In this paper, we present the extraction of specialized knowledge from a corpus of karstology literature. Domain terms are extracted by comparing the domain corpus to a reference corpus, and several heuristics to improve the extraction process are proposed (filtering based on nested terms, stopwords and fuzzy matching). We also use a word embedding model to extend the list of terms, and evaluate the potential of the approach from a term extraction perspective, as well as in terms of semantic relatedness. This step is followed by an automated term alignment and analysis of the Slovene and English karst terminology in terms of cognates. Finally, the corpus is used for extracting domain definitions, as well as triplets, where the latter can be considered as a potential resource for complementary knowledge-rich context extraction and visualization.

Keywords: karstology; term extraction; term embeddings; term alignment; definition extraction; triplets; specialized corpora

1. Introduction

The totality of means of expression in a language can be divided into general language and specialized language. Even if there is no distinct boundary between the two, it can be said that general language defines the sum of the means of linguistic expression encountered by most speakers of a given language, whereas specialized language goes beyond the general vocabulary based on the socio-linguistic or the subject-related aspect. The latter arises as a consequence of constant development and specialization in the fields of science, technology, and sociology (Svensen, 1993: p. 48-49). Similar to the definition of technical language by Svensen, in the context of terminology, specialized language, also called language for special purposes, is defined as a "language used in a subject field and characterized by the use of specific linguistic means of expression" (ISO 1087-1:2000).

If lexicologists and lexicographers mainly focus on words or lexemes, terminologists focus on terms, i.e., the words with a protected status when used in special subject domains (Pearson, 1998: p. 7). In contemporary approaches, the dichotomy 'word-term' is wiped-out. For Kageura (2002) terms are functional variants of words. Cabré Castellví (2003: p. 189) claims that all terms are words by nature and notes that "we recognize the terminological units from their meaning in a subject field, their internal structure and their lexical meaning". According to Myking (2007: p. 86), the traditional terminology is concept-based and the new directions are lexeme-based.

A definition is a characterization of the meaning of the lexeme (Jackson, 2002: p. 93). It is "a representation of a concept by a descriptive statement which serves to differentiate it from related concepts" (ISO 12620:2009). The concept to be defined is called a *definiendum*, the part defining its meaning *definiens*, and the optional element (usually a verb) connecting the two parts in a sentence, is called a hinge.

Granger (2012) highlights six most significant innovations of electronic lexicography in comparison to the traditional methods: a) corpus integration, meaning the inclusion of authentic texts in the dictionaries; b) more and better data, since there are no more space

limitations and one has the possibility to add multimedia data; c) efficiency of access (quick search and different possibility of database organization); d) customization, meaning that the content can be adapted to the user's needs; e) hybridization, denoting that the limits between different types of language resources—e.g., dictionaries, encyclopedias, term banks, lexical databases, translation tools—are breaking down; and f) user input, since collaborative or community-based input is integrated. Similar can be claimed for terminological work, where recent approaches in terminology science consider knowledge (represented in texts) as conceptually dynamic and linguistically varied (Cabré, 1999; Kageura, 2002), and where novel methods in data acquisition, organization and representation, are being constantly developed. Knowledge can be extracted from specialized resources automatically, benefiting from the advances in the field of natural language processing. Moreover, attempts in dynamic, visual representation of domain knowledge have been proposed in recent years, e.g., EcoLexicon[1] (Faber et al., 2016).

In this work, we present the extraction of specialized knowledge from a corpus of karstology, i.e. an interdisciplinary domain at the intersection of geology, hydrology, and speleology. The domain is of high interest, as karst is possibly the most prominent geographical feature of Slovenia (with karst formations being some of popular tourist and natural attractions in Slovenia). It is also an interesting example of how terminology is dynamically evolving in a cross-linguistic context. The literature published in English contains many local Slovenian scientific terms and toponyms for typical geomorphological karst structures, which makes it appropriate for research and identification of cognates, as well as homonym terms, with possible differences in meaning accross cultures.

Withing the TermFrame[2] project, we focus on the specialized knowledge of karst science, and plan to develop methods that allow for context- and language-dependent investigation into a domain, relying on semi-automated tools. In this paper, we apply some of the methods that we have previously developed to a new domain, resulting in a repository of karst term and definition candidates in Slovene and English, contributing to the karstology terminological science. Next, we propose a word embedding based term list extension and triplet extraction that can be used for visualization. These are novel components, contributing to terminological domain modelling.

This paper is structured as follows. After presenting the related work in automated specialized knowledge extraction in Section 2, we present the resources used (Section 3), methods (Section 4), results (Section 5) and conclude the paper with a discussion and plans for future work (Section 6).

2. Related work

Terminological work has undergone a significant change with the emergence of computational approaches resulting in semi-automated extraction of terms, definitions and other knowledge structures from raw text. Automatic terminology extraction has been implemented for various languages, including English (e.g., Sclano & Velardi, 2007; Frantzi & Ananiadou, 1999; Drouin, 2003) and Slovene (e.g., Vintar, 2010; Pollak et al., 2012), which are the languages in our corpus. In the last few years, word embeddings (Mikolov et al., 2013) have become a very popular natural language processing technique and several attempts have already been made to utilize word embeddings for terminology extraction

[1] http://ecolexicon.ugr.es/en/index.htm
[2] http://termframe.ff.uni-lj.si/

(e.g., Amjadian et al., 2016; Zhang et al., 2017). We use word embeddings techniques for extending term lists.

Also in bilingual term extraction and alignment, numerous approaches have been proposed, including Gaussier (1998), Kupiec (1993), Lefever et al. (2009), Vintar (2010), Baisa et al. (2015), as well as Aker et al. (2013), who treat bilingual term alignment as a binary classification task. The modified version of the latter approach described in Repar et al. (2018), is also used in this paper.

Automated definition extraction approaches have been developed for several languages, including English (e.g., Navigli & Velardi, 2010), Slovene (e.g., Fišer et al., 2010) and multilingual methods (e.g., Faralli & Navigli, 2013). In our work we use a pattern-based definition extraction method for English and Slovene (Pollak et al., 2012).

In addition to definitions, authors have focused on extracting different types of semantic relations. Pattern-based approaches (Hearst, 1992; Roller et al., 2018), as well as machine learning techniques have been proposed (cf. Nastase et al., 2013). In contrast to extracting predefined semantic relations, the Open Information Extraction (OIE) paradigm considers relations as expressed by parts of speech (Fader et al., 2011), paths in a syntactic parse tree (Ciaramita et al., 2005), or sequences of high-frequency words (Davidov & Rappoport, 2006). In our experiments we use the ReVerb triplet extractor by Etzioni et al. (2011).

This study presents the knowledge extraction steps within the TermFrame project, complementing previous work in karstology modelling presented in Vintar & Grčić-Simeunović (2017), and contributing to the emerging karstology knolwedge base. The extracted knowledge was used in the frame-based annotation approach, identifying the semantic categories, relations and relation definitors in definitions of karst concepts, as presented in Vintar et al. (2019), as well as in topic modeling using term co-occurrence network presented in Miljković et al. (2019). The work is also closely related to Faber et al. (2016), a multilingual visual thesaurus of environmental science, which was developed following frame-based, cognitively-oriented approach to terminology.

3. Resources

The corpus of karstology was constructed within the TermFrame project; it consists of Slovene, Croatian and English texts. We focus on the Slovene and English parts of the TermFrame corpus (v1.0). The English subcorpus contains cca. 1.6 M words and the Slovene one cca. 1 M words (see Table 1 for details).

Table 1: Statistics for English and Slovenian subcorpora.

	English	Slovene
Vocabulary size	64,079	73,813
Documents	24	60
Sentences	103,322	57,575
Words	1,673,132	1,041,475
Tokens	1,972,320	1,231,039
Type-to-token ratio	0.032	0.060

In addition, we are using a short gold standard list of Karst domain terms, called QUIKK termbase[3]. The QUIKK term base consists of terms in four languages, but for the purposes of our experiments, the Slovene and English term lists are used, containing 57 and 185 terms, respectively.

4. Methods

4.1 Term candidate extraction

First, we present the procedure of extracting terms by comparing the words in the noun phrases in the domain and reference corpora, and next we present a method using word embeddings to extend the list of terms.

4.1.1 Statistical term extraction

For extracting domain terms we use the LUIZ-CF term extractor (Pollak et al., 2012), which is a variant of LUIZ (Vintar, 2010) refined with scoring and ranking functions. The term extraction uses part-of-speech patterns for detecting noun phrases and compares the frequencies of words (lemmas) in the noun phrase in the domain corpus of karstology and a reference corpus.

The output is a list of term candidates in Slovene and English, above a selected frequency[4] and/or termhood threshold. In addition, we applied the following filtering and term merging procedures:

- *Nested terms filtering*: Nested terms are the terms that appear within other longer terms and may or may not appear by themselves in the corpus (Frantzi et al., 2000). As in Repar et al. (2019), the difference between a term and its nested term is defined by a frequency difference threshold: if a term in a corpus appears predominantly within a longer string, only the longer term is returned. If not (if a shorter term appears independently of a longer term more frequently than the set parameter), both terms are included in the final output.[5]
- *Stop words filtering*: If a term candidate is found on the stop word list, the term is excluded from the final list.[6]
- *Term merging by fuzzy matching:* Frequently, we can find terms that are extracted as separate terms but are in fact duplicates because they are written in different variants. This can be due to spelling variations (e.g., British and American English, using hyphenation or not), typos (which are relatively frequent when we deal with large text collections), errors due to pdf-to-text conversions etc. The proposed term merging is based on Levenshtein edit distance (Levenshtein, 1966): if two terms are nearly identical (default threshold is 95%), they will be merged and mapped to a

[3] http://islovar.ff.uni-lj.si/karst
[4] We set minimum frequency to 15.
[5] In our experiments, the parameter is set to 15 to match minimum frequency.
[6] General stop words are not problematic, as they are frequent also in a reference corpus, and therefore not identified as terms by LUIZ-CF. However, the words specific to the academic discourse, are not frequent in general language and therefore often appear as extracted term candidates. To exclude them, we use the following short stop words list: *example, use, source, method, approach, table, figure, percentage, et, al., km.*

common identifier. In addition, a rule which handles the case when two terms have a different prefix but the same tail and should not be recognized as duplicates can be applied.

4.1.2 Extending term lists with word embeddings

Word embeddings are vector representations of words, where each word is assigned a multi-dimensional vector of real numbers, characterizing the word based on the lexical context in which it appears. When vectors are computed on very large corpora, and especially with recent advances in models using neural networks, these representations have seen a huge success within various natural language processing tasks.

The embeddings capture certain degree of semantics, as words that are similar or semantically related are closer together in the vector space. Previous research conducted by Diaz et al. (2016) showed that embeddings can be successfully used for expanding queries on topic specific texts. In this research, we test if word embeddings can be used for a similar task of extending the gold standard term lists to find more domain terms. According to the research conducted by Diaz et al. (2016), embeddings trained only on small topic specific corpora outperform non-topic specific general embeddings trained on very large general corpora for the task of query expansion due to strong language use variation in specialized corpora. Therefore, we use the same approach for extending the term list and train custom embeddings on the specialized corpus instead of using pretrained embeddings.

In our experiments, we have trained FastText embeddings (Bojanowski et al., 2017) on the Slovenian and English karst subcorpora and use them to find twenty closest words (according to cosine distance between embeddings) for the first fifty terms in the QUIKK term base[7]. These related words are sorted according to their proximity to the term and the first, second, tenth and twentieth ranked words are used in manual evaluation. Embeddings for multi-word terms are generated by averaging the word embeddings for each word in the term.[8]

4.2 Cognates detection and term alignment

English terms are mapped to Slovene equivalents using a data mining approach by Aker et al. (2013) reimplemented in Repar et al. (2018). Bilingual term alignment is treated as a binary classification, with a support vector machine classifier trained on various dictionary and cognate-based features that express correspondences between the words (composing a term) in the target and source language. The first take advantage of dictionaries (Giza++) created from large parallel corpora, and the latter exploit string-based word similarity between languages (cf. Gaizauskas et al., 2012). In addition, the cognate-based features (see Table 2) allow to identify cognate term pairs, which are interesting as karst terms in

[7] To be exact, 50 English terms, and 47 Slovene terms, since only 47 Slovenian terms from the QUIKK term base appear in the Slovenian corpus.

[8] There are several possible multi-word term aggregation approaches, such as summation of component word vectors, averaging of component word vectors, creating multi-word term vectors, etc. As comparing different techniques is beyond the scope of this study, we decided for the simple averaging technique, as previous research on this topic conducted on the medical domain (Henry et al., 2018) found no statistically significant difference between any multi-word term aggregation method.

different languages clearly share their origin, but there exist also well-known examples of non-equivalent cognates (e.g., Slovene dolina vs. English doline).

Table 2: Cognate-based features used for term alignment.

Feature	Description
Longest Common Subsequence Ratio	Measures the longest common non-consecutive sequence of characters between two strings
Longest Common Substring Ratio	Measures the longest common consecutive string (LCST) of characters that two strings have in common
Dice similarity	2*LCST / (len(source) + len(target))
Normalized Levensthein distance (LD)	1 - LD / max(len(source), len(target))

4.3 Definition candidates extraction

We use the pattern-based module of the definition extractor (Pollak et al., 2012), which is available online.[9] The soft pattern matching is used to extract sentences of forms NP is NP, NP refers to NP, NP denotes NP, etc., and the parameters contain language (EN, SL), as well as the position of the term in Slovene (if the term must be at the beginning of the sentence, after a larger set of predefined start patterns (our choice) or anywhere in a sentence).

4.4 Triplet extraction

As predefined definition patterns (cf. Section 4.3) were designed for extracting specific knowledge contexts, we complement the approach by open-relation extraction (this experiment is conducted only for English, as for Slovene the tools are not available). We use ReVerb (Fader et al., 2011), which extracts relation phrases and their arguments and results in triplets of form:

```
<argument1, relation phrase, argument2>
```

We believe that in the case that `argument1` and `argument2` match domain terms, the triplets can be exploited as a method for extraction of knowledge-rich contexts (an alternative to definitions). They are also a useful input for visualization of terminological knowledge and can meet the needs of frame-based terminology, aiming at facilitating user knowledge acquisition through different types of multimodal and contextualized information, in order to respond to cognitive, communicative, and linguistic needs (Gil-Berrozpe et al., 2017). Previously, triplets have been used in other domains, e.g., in systems biology for building networks from domain literature (Miljković et al., 2012).

[9] http://clowdflows.org/workflow/8165/

5. Evaluation setting and results

5.1 Term candidates extraction

5.1.1 Statistical term extraction

We extracted 4397 English term candidates and 2946 Slovene term candidates. A domain expert and a linguist specialized in terminology with high domain understanding manually evaluated all term candidates for Slovene and the top 1823 (above a selected threshold)[10] term candidates for English. The following categories were used:

- Not a term (label: 0)
- Karst term (label: 1)
- Broader domain terms (label: 2)
- Named entity (label: 3)

To distinguish between karst and broader domain terms, the following criterium is used. While karstology is in itself an interdisciplinary field, in TermFrame the focus is on karst geomorphology entailing surface and underground landforms, and karst hydrology with its typical forms and processes. Terms from neighbouring domains (geography, biology, geochemistry etc.) which are not exclusive to karst are considered broader domain terms. In case of disagreement, the two annotators made consensus on the final category. As presented in Table 3, the resulting list of terms contains 351 karst terms for English and 158 for Slovene. The newly extracted karst terms, such as *cave, uvala, doline, denudation* describing landforms, processes, environment, etc., can serve for the extension of the manual QUIKK karstology term base, while for example the term candidate *karst region* is not considered a term because it is too generic and compositional, denoting a different underlying semantic relation (a region which contains karst).

The precision of term extraction is 0.516 for English and 0.235 for Slovene. For examples of terms in each category, see Table 4, while top terms sorted by termhood score for English and Slovene are presented in Tables 5 and 6, respectively.

Table 3: Term extraction results. Precision is calculated as the sum of all three positive categories (1, 2, 3) divided by the number of evaluated terms.

Lang	Evaluated terms	Not a term	Karst term	Broader domain term	Named entity	Precision
Slovene	2946	2228	158	194	341	0.235
English	1823	882	351	434	156	0.516

In addition, we evaluate our filtering methods. All nested terms (306 for English, 105 for Slovene) removed by the nested term filtering are correctly eliminated, the stop words

[10] The reason for the discrepancy in the number of evaluated terms is that the evaluation for Slovene yielded a much lower number of terms (categories 1 or 2) in Slovene than in English. Since we need a large number of terms for additional steps, i.e. term alignment, we instructed the evaluators to process the full list of term candidates for Slovene. If we took the same number of top terms for Slovene as for English (top 1823), we get the following results (cf. Table 3): Not a term: 1187, Karst term: 140, Domain term: 174, Named entity: 220, Precision: 0.293.

Table 4: Examples of term extraction evaluation categories.

Lang	Not a term	Karst term	Broader domain term	Named entity
Slovene	dinarska smer	slepa dolina	naplavna ravnica	Planinsko polje
	ilovnat material	udornica	mehansko preperevanje	Podgorski kras
	kataster jam	kalcijev karbonat	ravnovesna meja	Gorski kotar
English	deepest cave	karst aquifer	sea level	Southeast Asia
	world heritage	subterranean water	carbonic acid	Castleguard Cave
	largest spring	phreatic cave	cave habitat	Central America

Table 5: Top 20 English karst term candidates with frequencies and categorization to karst terminology (1), broader domain terminology (2), named entity (3) or non-term (0).

Rank	Frequency	Term	Categorization
1	19269	cave	1
2	451	karst aquifer	1
3	522	karst area	1
4	459	cave system	1
5	314	dinaric karst	3
6	414	carbonate rock	1
7	348	cave passage	1
8	218	crna reka	3
9	271	karst system	1
10	209	karst feature	1
11	192	karst terrain	1
12	201	karst landscape	1
13	203	karst region	0
14	192	karst spring	1
15	564	united state	3
16	146	troglobitic specie	2
17	187	cave entrance	1
18	227	lava tube	2
19	169	cave sediment	1
20	164	karst rock	1

filter did not detect any terms which should not be removed, and all near duplicates (11 for English, 22 for Slovene) detected with the fuzzy match filter are also correct (e.g., “ground-water” was detected as a duplicate of “ground water”).

5.1.2 Extending term lists with word embeddings

The method was tested on 47 English and 50 Slovene source terms (i.e. the terms from the gold standard list), for which out of 20 most related words (according to the cosine distance between the source term and the related word), four per each source term were selected for evaluation (first, second, tenth and twentieth ranked words), resulting in 200 term-word pairs for English and 188 for Slovene.[11] Examples of ranked related words for five English and five Slovene terms are presented in Table 7.

[11] In this section, we intentionally name related words as words and not as terms, to contrast them to the gold standard list of terms to which they are compared. As shown in the evaluation, they can be in next step evaluated as terms or not.

Table 6: Top 20 Slovene karst term candidates with frequencies and categorization to karst terminology (1), broader domain terminology (2), named entity (3) or non-term (0).

Rank	Frequency	Term	Categorization
1	1966	nadmorska višina	0
2	9543	jama	1
3	4472	kras	1
4	6359	voda	0
5	713	slepa dolina	1
6	4481	dolina	0
7	405	brezstropa jama	1
8	2948	apnenec	1
9	623	Pivška kotlina	3
10	2573	sediment	0
11	3418	dno	0
12	425	erozijski jarek	2
13	3608	polje	1
14	2770	rov	1
15	728	kraško polje	1
16	2049	udornica	1
17	4619	del	0
18	2564	kamnina	2
19	507	suha dolina	1
20	3882	oblika	0

Table 7: Examples of ranked related words for five English (upper five examples) and five Slovene (lower five examples) terms.

Term	R1	R2	R10	R20
sinkhole	shakehole	suburban	sinkpoint	dump
aggressive water	aggressively	aggressiveness	qc	coldwater
epikarst zone	epikarstic	subcutaneous	cutaneous	epiphreatic
caprock sinkhole	sinkpoint	overbank	suburb	evacuation
seacave	seacoast	sealevel	vrulja	caveand
udornica	udornina	zapornica	koliševka	kamojstrnik
agresivna voda	sposoben	mehurček	skoznjo	preniči
epikras	epikraški	prenikujoč	epr	vadozen
vrtača	vrtačast	mikrovrtača	globel	neizravnan
rečna jama	reža	narečen	mohoričev	vodokazen

The two human evaluators evaluated the related words according to two criteria:

- Is the word a term
- Semantic similarity to the term

The first criterion is measured on a scale with four nominal classes (see Section 5.1.1), while the second criterion uses a numerical scale from zero to ten, following the evaluation procedure of Finkelstein et al. (2002), where zero suggests no semantic similarity and ten suggests very close semantic relation (fractional scores were also allowed). The inter-annotator agreement between two evaluators (according to the Cohen's kappa coefficient)

is 0.689 for the first criterion and 0.513 for the second criterion for English and 0.594 for the first criterion and 0.389 for the second criterion for the Slovene evaluation.

Table 8 presents results for the evaluation of embeddings-based term extension. Out of 200 English term-word pairs, 112 were manually labeled as term-term pairs by at least one evaluator which suggests that, at least for English, embeddings can be used for extending the term list. Out of these 112 related terms, 52 were labeled as karst specific terms by at least one evaluator. For Slovenian, the results are worse, since out of 188 term-word pairs only 69 were labeled as term-term pairs and out of these only 36 are karst specific.

Out of 112 English term-term pairs, 62 were ranked first and second and 50 were ranked tenth and twentieth according to the cosine distance similarity. Out of 69 Slovenian term-term pairs, 39 were ranked first or second and 30 were ranked as tenth or twentieth. This suggests that words that have most similar embeddings to terms according to the cosine distance (rank 1 and rank 2) are also more likely to be terms themselves than words that have less similar embeddings (rank 10 and rank 20). Similar applies to karst specific term-term pairs, where for English 30 were ranked first or second and 22 were ranked tenth or twentieth. For Slovenian, 24 out of 36 were ranked first or second and 12 were ranked tenth or twentieth.

When it comes to semantic similarity, unsurprisingly better ranked related words were manually evaluated as semantically more similar. For example, the first ranked (most similar to terms according to the cosine distance) English related words got an average semantic similarity score[12] of 4.040 out of ten and first ranked Slovenian related words got an average semantic similarity score of 4.468. These are larger averages than semantic similarity score averages of 2.610 and 3.064 for English and Slovenian related words ranked as twentieth. Another interesting observation is the fact that the average semantic similarity score is the largest for English karst specific term-terms pairs (5.702) and much lower if all the term-word pairs are considered (3.325). If we consider all term-term pairs, the average semantic similarity score is 4.710. Same applies for Slovenian term-word pairs, with semantic similarity score average rising from 3.859, when all term-words pairs are considered, to 5.536, when only term-term pairs are considered, and up to 6.722, when only karst specific term-term pairs are considered.

We also measure correlation between cosine distances and the semantic similarity scores for term-word pairs using Pearson and Spearman correlation coeficients. The correlation is generally low, the highest correlation being measured for Slovenian Karst specific term-term pairs where the Pearson correlation reached the value of 0.341 and Spearman the value of 0.208. There was no correlation measured on Slovene term-term pairs and surprisingly, a small negative Pearson correlation was measured on Slovenian karst specific term-term pairs and a small negative Spearman correlation was measured on English pairs which were labeled as terms.

5.2 Cognate detection and term alignment

We evaluate the approach first on the QUIKK gold standard, where 100% precision and the recall above 40% were obtained. Next, we add to the QUIKK gold standard also

[12] Semantic similarity score for each related word is calculated as an average between the two semantic similarity scores given by two evaluators.

Table 8: English and Slovenian embeddings evaluation according to two criteria described in Section 4.1.2. *Avg. sem. score* stands for the average of manually prescribed semantic similarity scores for each term-word pair, *Avg. cos. dist* stands for the average cosine distance, *Pearson corr.* is a Pearson correlation coefficient between the semantic similarity score and cosine distance values and *Spearman corr.* is a Spearman correlation coefficient between the semantic similarity score and cosine distance values.

	English				Slovene			
All words	200				188			
Avg. sem. score	3.325				3.859			
Avg. cos. dist.	0.747				0.760			
Pearson corr.	0.181				0.231			
Spearman corr.	0.136				0.194			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	50	50	50	50	47	47	47	47
Avg. sem. score	4.040	3.540	3.110	2.610	4.872	4.468	3.032	3.064
Terms	112				69			
Avg. sem. score	4.710				5.536			
Avg. cos. dist.	0.757				0.771			
Pearson corr.	0.176				-0.018			
Spearman corr.	0.160				-0.016			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	32	30	29	21	17	22	15	15
Karst terms	52				36			
Avg. sem. score	5.702				6.722			
Avg. cos. dist.	0.761				0.780			
Pearson corr.	0.151				-0.152			
Spearman corr.	0.070				-0.067			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	16	14	15	7	12	12	5	7
Not Terms	88				119			
Avg. sem. score	1.563				2.887			
Avg. cos. dist.	0.734				0.753			
Pearson corr.	-0.010				0.341			
Spearman corr.	-0.110				0.208			
	R1	R2	R10	R20	R1	R2	R10	R20
Distribution	18	20	21	29	30	25	32	32

the terms extracted using statistical method and term embeddings that were positively evaluated. The total list of 908 English terms and 391 Slovene terms were input to the term alignment algorithm. The resulting list of 93 aligned term pairs was manually evaluated. In this experiment, the precision was 77.42% (72 term alignments out of 93 were correct), while the recall could not be calculated, as the gold standard alignment was not available.

As described in Section 4.2, karst terminology contains a considerable amount of cognates. See Table 9 for cognate values for Longest Common Substring Ratio, Longest Common Subsequence Ratio, Dice Similarity, and Normalized Levensthein Distance).

5.3 Definition candidates extraction

In total, 1320 definition candidates were extracted for English, and 1218 for Slovene. Definition candidates were manually validated by domain experts following two criteria:

whether the sentence defines the concept, and whether the concept belongs to the domain of karstology. To distinguish between definitions and non-definitions the experts checked whether the sentence explains what the concept is, either by specifying its hypernym and a set of distinguishing features (analytical), or by listing its hyponyms (extensional), or by using another explanatory strategy (e.g., functional definitions). The definition candidates were then assigned one of the following three categories:

- Definitions of karst terms (Example: *Aggressiveness is an attribute of groundwater that corresponds to a chemical potential for mobilization of a dissolved matter from the rock.*)
- Definitions of broader domain terms (biology, geology etc.). (Example: *Exploration geophysics is the science of seeing into the earth without digging or drilling.*)
- Non-definitions (Example: *The oldest rocks are the sandstones of Permian age, which are only locally present.*)

Table 9: Cognate scores for a sample of Slovene and English term pairs.

English term	Slovene term	LCSTR	LCSSR	Dice	NormLD
mineralization	mineralizacija	0.71	0.79	0.71	0.79
salinization	salinizacija	0.67	0.75	0.67	0.75
nitrification	nitrifikacija	0.54	0.69	0.54	0.69
aggressive water	agresivna voda	0.25	0.63	0.27	0.50
karst plateau	kraška planota	0.27	0.60	0.29	0.40
karst	kras	0.20	0.60	0.22	0.40
marble	marmor	0.50	0.50	0.50	0.50
karst drainage	kraška drenaža	0.19	0.50	0.20	0.38
karst phenomena	kraški pojav	0.13	0.47	0.14	0.20
linear stream cave	linearna epifreatična jama	0.22	0.44	0.27	0.44

Table 10: Number of extracted definition candidates, evaluated as karst definitions, broader domain definitions and non-definitions.

	English	Slovene
Karst definitions	218	260
Broader domain definitions	187	166
Non definitions	915	792
All definition candidates	1320	1218

As presented in Table 10, for English, out of 1320 definition candidates, 218 were evaluated as karst definitions, and additional 187 as broader domain definitions (the precision of the definition extraction on karst domain is thus 0.16 for strictly karst domain definitions, and 0.31 for broader domain definitions (incl. karst definitions). For Slovene, there are 1218 definition candidates, out of which 260 are karst definitions and 166 are from broader domain. The precision for definition extraction for Slovene is thus 0.21 for strictly karst domain, and 0.35 for karst and broader domain.

The karst definitions were then used by domain experts and linguists in the scope of the TermFrame project for fine-grained, annotation process, following frame-based ter-

minology principles (Faber, 2015). The annotation principles and results are presented in Vintar et al. (2019), where several annotation layers are proposed: definition element layers (definiendum, definitor and genus); semantic categories (top level concepts are are landforms, processes, geomes, entities, instruments/methods) and relations (16 relations, such as has_form, has_cause).

5.4 Triplet extraction

The English subcorpus yielded 80,564 triplets. Below we list selected examples of relevant triplets that are closely related to the karst domain:

- `<Karst areas, commonly lack, surface water>`
- `<Karst areas, have, numerous stream beds that are dry except during periods of high runoff>`
- `<Sinkholes located miles away from rivers, can flood, homes and businesses>`
- `<Karst areas, offer, important resources>`
- `<Some collapse sinkholes, develop, where collapse of the cave roof reaches the surface of the Earth>`

The extracted triplets are analysed according to the most common relation patterns, to estimate their potential for extending predefined definition patterns. From the relation phrase part of the triplet, the verb is identified, showing the most frequent verb structures. We remove all stopwords from the relation phrase using a general list of 174 English stopwords. Table 11 lists 20 most frequent verb structures found in the processed 24 documents. The results show that many karst-specific relations can be detected (e.g., verbs related to different geological processes, such as *occur*, *develop* and *form*) but still many general verbs are also frequent. The frequent relations from triplets will be discussed in relation to the predefined set of relations used in definition frames annotation (cf. Vintar et al., 2019).

	verb	count		verb	count
1	found	1451	11	appear	336
2	occur	1347	12	consist	323
3	use	878	13	represent	321
4	form	811	14	locate	313
5	develop	787	15	include	312
6	know	646	16	contain	310
7	provide	528	17	made	306
8	show	428	18	result	295
9	take	397	19	depend	273
10	describe	337	20	extend	272

Table 11: 20 most frequent verb structures compiled from 80,564 triplets. Note that stopwords were removed from verb structures.

For visualization, after filtering the triplets by keeping only the ones where in a triplet `<argument1, relation phrase, argument2>` the two arguments are karst terms[13], we

[13] QUIKK terms and manually evaluated terms from Section 5.1.1.

construct a network where arguments are used as nodes and relation phrases as arcs. A visualisation of a part of the triplet network obtained using Biomine network visualisation tool (Eronen & Toivonen, 2012) is shown in Figure 1.

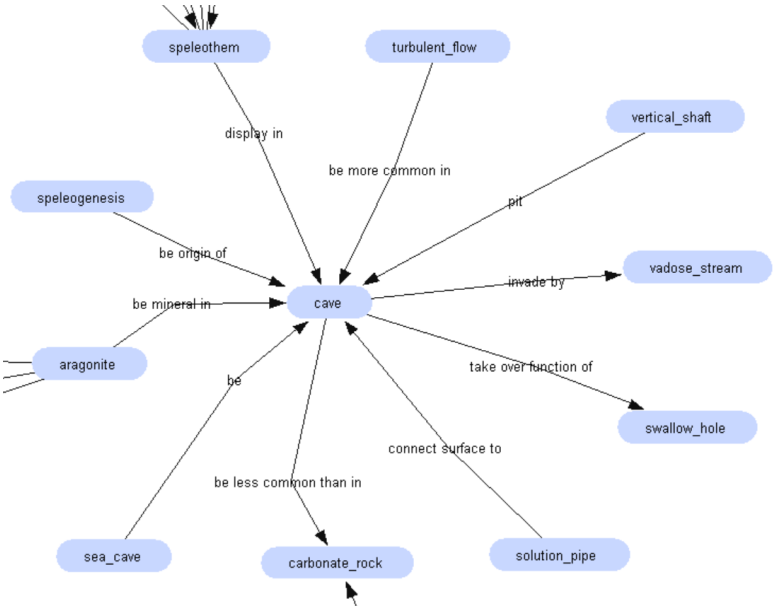

Figure 1: Visualisation of a part of the triplet network. Prior to the visualization, relation phrases were lemmatised and the triplets were filtered according to the short gold standard list of Karst domain extended with an additional evaluated list of terms.

6. Conclusion and further work

We model domain knowledge utilizing a range of natural language processing techniques, including term extraction (using statistical methods, filtering and word embeddings), term alignment and cognates detection, definition extraction and triplet extraction. The proposed techniques form a pipeline for contemporary terminological work, relying on semi-automated processes for knowledge extraction from specialized domain corpora. Several modules in the pipeline rely on existing techniques, which were refined for the purposes of this work (e.g., term extraction), while we believe that the use of embeddings and triplets has not yet been sufficiently explored in the context of lexicography and terminography. The hypothesis was that embeddings offer not only a possibility of extending a list of terms, but also for grouping them to semantically related concepts, which can be of great value in the organization of domain knowledge (in term bases and similar resources), and also in contemporary lexicography resources.

We apply the proposed pipeline to a corpus of karst specialized texts. The main value of the evaluation steps of term and definition extraction is to obtain new gold standard karst knowledge resources that will be used in the scope of the TermFrame project for fine-grained analysis and novel visual representation corresponding to the cognitive shifts in recent terminology science approaches. On the other hand, we believe that the evaluation

of word embeddings opens new perspectives to e-lexicography and terminography, as it shows that popular techniques from natural language processing are relatively successful for automatically extending the gold standard term lists (cca. half of English and one third of Slovene terms being valid terms). The evaluation also shows that the semantic similarity score is higher for the closest matching words (considering cosine similarity between embeddings) than for the lower ranked words, which suggests that embeddings do in fact manage to capture some semantic relations despite a relatively small training corpus. On the other hand, correlation between cosine similarity and manual similarity score is weak, which might indicate high variance in cosine similarity for related words for different terms. We believe that semantic information has a huge potential for contributing to the organization of term bases and visually interesting knowledge maps. In the same line, we illustrate how triplet extraction in combination with term matching can serve as a knowledge representation module used for visualizaton.

In future work, we will consider extending the corpus by using webcrawling techniques. Next, our aim is to merge the pipeline to a set of services to support users in a knowledge extraction process, for populating term bases, as well as in knowledge visualisation. We believe that such tools will contribute to better understanding of similarities and differences in terminological expression between languages, and support representations reflecting dynamic culture and language specific knowledge.

7. Acknowledgements

The work was supported by the Slovenian Research Agency through core research programme (P2-0103) and research project Terminology and knowledge frames across languages (J6-9372). This work was supported also by the EU Horizon 2020 research and innovation programme, Grant No. 825153, EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflect only the authors' views and the EC is not responsible for any use that may be made of the information it contains. We would also like to thank Š. Vintar, U. Stepišnik, D. Miljković and other members of the TermFrame project for their collaboration.

8. References

Aker, A., Paramita, M. & Gaizauskas, R. (2013). Extracting bilingual terminologies from comparable corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1. pp. 402–411.

Amjadian, E., Inkpen, D., Paribakht, T. & Faez, F. (2016). Local-Global Vectors to Improve Unigram Terminology Extraction. In *Proceedings of the 5th International Workshop on Computational Terminology (Computerm2016)*. pp. 2–11.

Baisa, V., Ulipová, B. & Cukr, M. (2015). Bilingual terminology extraction in Sketch Engine. In *9th Workshop on Recent Advances in Slavonic Natural Language Processing*. pp. 61–67.

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, pp. 135–146.

Cabré, M.T. (1999). *Terminology: Theory, Methods, and Application*. Amsterdam, The Netherlands and Philadelphia, USA: John Benjamins Publishing.

Cabré Castellví, M.T. (2003). Theories of Terminology: Their Description, Prescription and Explanation. *Terminology 9 (2)*, p. 163–199.

Ciaramita, M., Gangemi, A., Ratsch, E., Šaric, J. & Rojas, I. (2005). Unsupervised Learning of Semantic Relations Between Concepts of a Molecular Biology Ontology. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*. pp. 659–664.

Davidov, D. & Rappoport, A. (2006). Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words. In *Proceedings of the 21th International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia, pp. 297–304.

Diaz, F., Mitra, B. & Craswell, N. (2016). Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, p. 367–377.

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1), pp. 99–117.

Eronen, L. & Toivonen, H. (2012). Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC Bioinformatics*, 13(1), pp. 1–21.

Etzioni, O., Fader, A., Christensen, J., Soderland, S. & Mausam (2011). Open Information Extraction: The Second Generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume One (IJCAI'11)*. Barcelona, Catalonia, Spain, pp. 3–10.

Faber, P. (2015). Frames as a framework for terminology. In H. Kockaert & F. Steurs (eds.) *Handbook of Terminology*. John Benjamins, p. 14–33.

Faber, P., León-Araúz, P. & Reimerink, A. (2016). EcoLexicon: new features and challenges. In *Proceedings of GLOBALEX 2016: Lexicographic Resources for Human Language Technology in conjunction with the 10th edition of the Language Resources and Evaluation Conference*. pp. 73–80.

Fader, A., Soderland, S. & Etzioni, O. (2011). Identifying Relations for Open Information Extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*. Edinburgh, United Kingdom1: Association for Computational Linguistics, pp. 1535–1545.

Faralli, S. & Navigli, R. (2013). A Java Framework for Multilingual Definition and Hypernym Extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 103–108. URL https://www.aclweb.org/anthology/P13-4018.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1), pp. 116–131.

Fišer, D., Pollak, S. & Vintar, Š. (2010). Learning to Mine Definitions from Slovene Structured and Unstructured Knowledge-Rich Resources. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta, pp. 2932–2936.

Frantzi, K., Ananiadou, S. & Mima, H. (2000). Automatic recognition of multi-word terms:. the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2), pp. 115–130.

Frantzi, K.T. & Ananiadou, S. (1999). The C-Value/NC-Value Domain Independent Method for Multi-Word Term Extraction. *Journal of Natural Language Processing*, 6(3), pp. 145–179.

Gaizauskas, R., Aker, A. & Yang Feng, R. (2012). Automatic bilingual phrase extraction from comparable corpora. In *24th International Conference on Computational Linguistics*. p. 23.

Gaussier, E. (1998). Flow Network Models for Word Alignment and Terminology Extraction From Bilingual Corpora. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (Coling-ACL)*. pp. 444–450.

Gil-Berrozpe, J., León-Araúz, P. & Faber, P. (2017). Specifying Hyponymy Subtypes and Knowledge Patterns: A Corpus-based Study. In *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference*. pp. 63–92.

Granger, S. (2012). Electronic Lexicography-from Challenge to Opportunity. In S. Granger & M. Pacqot (eds.) *Electronic Lexicography*, chapter Introduction. Oxford University Press, p. 1–15.

Hearst, M.A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2 (COLING'92)*. pp. 539–545.

Henry, S., Cuffy, C. & McInnes, B.T. (2018). Vector representations of multi-word terms for semantic relatedness. *Journal of biomedical informatics*, 77, pp. 111–119.

ISO 1087-1:2000 (2000). International Standard: Terminology Work — Vocabulary — Part 1: Theory and Application. Standard cited from the Glossary of Terminology Management of DG TRAD – Terminology Coordination Unit of European Parliament (Last accessed June 17, 2019). Standard. URL http://termcoord.wordpress.com/glossaries/glossary-of-terminology-management/.

ISO 12620:2009 (2009). International Standard. Terminology and Other Language and Content Resources — Specification of Data Categories and Management of a Data Category Registry for Language Resources. Standard cited from ISOcat Web Interface (Last accessed December 1, 2013). Standard. URL https://catalog.clarin.eu/isocat/interface/index.html.

Jackson, H. (2002). *Lexicography: An Introduction*. Routledge.

Kageura, K. (2002). *The Dynamics of Terminology: A Descriptive Theory of Term Formation and Terminological Growth*. John Benjamins Publishing.

Kupiec, J. (1993). An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*. Columbus, OH.

Lefever, E., Macken, L. & Hoste, V. (2009). Language-Independent Bilingual Terminology Extraction from a Multilingual Parallel Corpus. In *Proceedings of the 12th Conference of the European Chapter of the ACL*. pp. 496–504.

Levenshtein, V.I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, p. 707.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings to The International Conference on Learning Representations 2013*.

Miljković, D., Kralj, J., Stepišnik, U. & Pollak, S. (2019). Communities of related terms in Karst terminology co-occurrence network. In *Proceedings of eLex 2019*.

Miljković, D., Stare, T., Mozetič, I., Podpečan, V., Petek, M., Witek, K., Dermastia, M., Lavrač, N. & Gruden, K. (2012). Signalling Network Construction for Modelling Plant Defence Response. *PLOS ONE*, 7(12), pp. 1–18. URL https://doi.org/10.1371/journal.pone.0051822.

Myking, J. (2007). No Fixed Boundaries. In A. Bassey (ed.) *Indeterminacy in Terminology and LSP: Studies in Honour of Heribert Picht*, chapter 6. Amsterdam, The Netherlands and Philadelphia, USA: John Benjamins Publishing, p. 73–91.

Nastase, V., Nakov, P., Séaghdha, D.Ó. & Szpakowicz, S. (2013). Semantic Relations Between Nominals. In G. Hirst (ed.) *Synthesis Lectures on Human Language Technologies*. London: Morgan & Claypool Publishers, pp. 1–119.

Navigli, R. & Velardi, P. (2010). Learning Word-Class Lattices for Definition and Hypernym Extraction. In *Proceedings of the Forty-Eighth Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, pp. 1318–1327.

Pearson, J. (1998). Terms in Context. In E. Tognini-Bonelli & W. Teubert (eds.) *SCL Series, Vol. 1*. Amsterdam, The Netherlands and Philadelphia, USA: John Benjamins Publishing.

Pollak, S., Vavpetič, A., Kranjc, J., Lavrač, N. & Špela Vintar (2012). NLP workflow for on-line definition extraction from English and Slovene text corpora. In J. Jancsary (ed.) *Proceedings of KONVENS 2012*. ÖGAI, pp. 53–60. Main track: oral presentations.

Repar, A., Martinc, M. & Pollak, S. (2018). Machine Learning Approach to Bilingual Terminology Alignment: Reimplementation and Adaptation. In A. onio Branco, N. Calzolari & K. Choukri (eds.) *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Paris, France: European Language Resources Association (ELRA).

Repar, A., Podpečan, V., Vavpetič, A., Lavrač, N. & Pollak, S. (2019). TermEnsembler: An ensemble learning approach to bilingual term extraction and alignment. *Terminology*, 25(1).

Roller, S., Kiela, D. & Nickel, M. (2018). Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 358–363. URL https://www.aclweb.org/anthology/P18-2057.

Sclano, F. & Velardi, P. (2007). TermExtractor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities. In *Proceedings of the 9th Conf on Terminology and Artificial Intelligence TIA 2007*. pp. 8–9.

Svensen, B. (1993). *Practical Lexicography: Principles and Methods Of Dictionary Making*. Oxford University Press.

Vintar, Š. (2010). Bilingual term recognition revisited: The bag-of-equivalents term alignment approach and its evaluation. *Terminology*, 16(2), pp. 141–158.

Vintar, Š. & Grčić-Simeunović, L. (2017). Definition frames as language-dependent models of knowledge transfer. *Fachsprache : internationale Zeitschrift für Fachsprachenforschung, -didaktik und Terminologie*, 39(1/2), pp. 43–58.

Vintar, Š., Saksida, A., Stepišnik, U. & Vrtovec, K. (2019). Knowledge frames in karstology: the TermFrame approach to extract knowledge structures from definitions. In *Proceedings of eLex 2019*.

Zhang, Z., Gao, J. & Ciravegna, F. (2017). SemRe-Rank: Incorporating Semantic Relatedness to Improve Automatic Term Extraction Using Personalized PageRank. *arXiv preprint arXiv:1711.03373*.