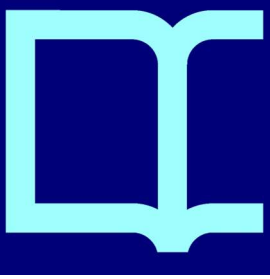

Noname manuscript No.
(will be inserted by the editor)

A system for terminology extraction and translation equivalent detection in real time

Efficient use of Statistical Machine Translation phrase tables

Antoni Oliver

Received: date / Accepted: date

Abstract In this paper we present a system for automatic terminology extraction and automatic detection of the equivalent terms in the target language to be used along with a Computer Assisted Translation (CAT) tool that provides term candidates and their translations in an automatic way each time the translator goes from one segment to the next one. The system uses several sources of information: the text from the segment being translated and from the whole translation project, the translation memories assigned to the project and a translation phrase table from a statistical machine translation system. It also uses the terminological database assigned to the project in order to avoid presenting already known terms. The use of translation phrase tables allows us to use very large parallel corpora in a very efficient way. We have used Moses to calculate and to consult the translation phrase tables. The program is written in Python and it can be used with any CAT Tool. In ours experiments we have used OmegaT, a well-known open source CAT tool. Evaluation results for English-Spanish and for three subjects (politics, finance and medicine) are presented.

Keywords automatic terminology extration · translation equivalent detection · translation memories · SMT translation models · computer assited translation

1 Introduction

Translation memories and term bases are the two main resources in a Computer Assisted Translation (CAT) tool. A translation memory is a store of

Antoni Oliver
Av. Tibidabo 39-43 - 08024 Barcelona - Catalonia (Spain)
Tel.: +34-932 542 187
Fax: +34-934 176 495
E-mail: aoliverg@uoc.edu

text segments in one language with their translations in other languages. Term bases are systemized collections of terminology that are generally (and nowadays always) found in computerized format.

In specialized translation the consistency on the translation equivalent selection for source language terms is very important. As many terms present in the translation project are not in the available term bases, an automatic method for source term detection and automatic selection of translation equivalents would be of great help for translators and language service providers. There are several methodologies that allow to automatically detect a set of term candidates from texts. These techniques are known as *Automatic Terminology Extraction* (ATE), *Terminology Extraction*, *Terminology Mining*, *Term Recognition*, *Glossary Extraction*, *Term Identification* and *Term Acquisition* [Heylen-2015]. There are also techniques that allow us to guess the translation equivalent of a term from a parallel corpus (or translation memory). When we try to automatically detect terms and their translation equivalents, we are talking about *Bilingual Terminology Extraction* [Gaussier-2001].

When a translation project is created, the project manager or the translator assigns these resources to the project. It is also very interesting to perform automatic terminology extraction on the texts to be translated, select the most relevant term candidates and perform an automatic selection of translation equivalents of these term in the available translation memories or parallel corpora. In this way, a specific term base for the current project can be created, and the translator will save time in terminological queries to external resources. This step, however, is not always performed.

In this paper we present an extension to a CAT tool that performs automatic bilingual terminology extraction each time the translator goes from one segment to the next one. In this way, the translator is presented with some term candidates in the source language, along with their translation equivalents in the target language. These new terms are not known in advance and they don't come from an existing term base.

The program performs automatic terminology extraction using well-known techniques and performs automatic detection of the equivalent terms in the target language searching in phrase tables from a statistical machine translation system. The use of phrase tables for bilingual terminology extraction has been previously used [Ideue-2011], but in previous works small parallel corpora have been used for creating the phrase tables. Our system performs a very efficient search in phrase tables allowing the use of very big phrase tables created from very large parallel corpora in nearly real time. A fast response time is important, as the system is intended to work along a Computer Assisted Translation Tool, and user should feel an immediate response when he finishes translating a segment and moves to the next one.

The paper is organized as follows: in the next section we briefly expose the main concepts related with CAT tools and describe the main characteristics of OmegaT, the tool we've used in our experiments. Then the main techniques for automatic terminology extraction, both monolingual and bilingual, are presented. The paper follows with a short description of translation memories,

and the very similar concept of parallel corpus as well as an explanation about the translation models we used in the system. In the next section a detailed description of our system is presented. Then we present our experiments and evaluation results. The paper finishes with some conclusions and future work we plan to perform on the system.

2 Computer-Assisted Translation Tools

Computer-assisted translation tools (or computer-aided translation tools) encompass a series of applications specially designed to efficiently provide translators with support in their work.

In a broader sense, the term computer-assisted translation covers all IT applications designed for tasks that are specific to the translation process. In a narrower sense, computer-assisted translation is a term generally used to refer to programs that aid translators by checking the contents of one or more translation memories, and also give the option of checking terminological glossaries. The importance that the translation memory concept holds is such that computer-assisted translation has often been referred to as translation memories or translation memory management systems.

There are a lot of CAT Tools in the market, some of them holding a free license. A recent survey [Eckl-2014] points out that the most used CAT Tools are: Trados (34.15%), WordFast (20.56%), MemoQ (11.85%), OmegaT (9.06%), Star Transit (2.79%) and XTM (2.09%). There is a wide range of users (19.50%) that uses other CAT Tools. Traditionally CAT Tools were programs that worked in the translator's computer. In recent years, cloud-based CAT tools are becoming popular, as XTM, Matecat or Memsource.

OmegaT[1] [Carretero-2010] is a mature CAT tool with a free license (GNU GPL) and a solid development project. This tool is widely used in translator training [Canovas-2011] and in professional environments [Eckl-2014]. Among a full set of interesting features, we can highlight two that facilitate the integration of our system:

- The translation projects are organized as a set of folders each of them including some components: the source files (folder *source*), the target files (folder *target*), the translation memories (folder *tm*), the term bases (folder *glossary*) and so on. As OmegaT uses standard formats (such as TMX for translation memories and TBX for term bases) it's easy to access to all the information about the project we are working with.
- The tool includes a system folder (*script* folder) where three files are located: *source.txt*, containing the source text of the current segment; *target.txt*, containing the target text of the current segment and *selection.txt*, containing the text highlighted by the user. Each time the translator goes from one segment to the next one, the content of the *source.txt* file changes.

[1] www.omegat.org

The system detects this change and reads this file to get the content of the new source segment text.

3 Automatic Terminology Extraction

Terminology is very important for translators but the way they address to this knowledge field is very different from terminologist. As stated by [Cabre-2010], from the point of view of translation, terminology is considered a tool to solve particular problems. For translators, terminology is *all the words I don't know and I need to find out, the words not found in a dictionary or the latest jargon* [Bononno-2000]. In this sense, interesting units for translators include not only the units that would be considered as *real terms* by a terminologist, but also other units requiring a specific translation in the given subject field.

As a CAT tool is a system for aiding translator in their daily work, and as translators think about terminology from a very practical point of view, a system as the one we are presenting will be of great help, as the translator will be presented with unknown terms and their translations in a fully automatic way.

Several other studies has addressed the importance of the identification and integration of bilingual domain-specific terms into Machine Translation systems, as for example [Arcan-2014] where the authors incorporated such a system into a Statistical Machine Translation system with the aim of increasing translation quality of high-specific texts in a CAT environment.

The methods for ATE can be classified in two main groups [Pazienza-2005]:

- Statistical methods: term extraction is performed based on statistical properties [Salton-1975] and usually implies the calculation of n-grams of words and filtering them with a list of stop-words. Although the most common and easiest to implement characteristic is simply the term candidate frequency, a long set of statistical measures and other approaches have been created for term candidate scoring and ranking [Astrakhantsev-2015].
- Linguistic methods [Bourigault-1992]: term extraction is performed based on linguistic properties. Most of the systems use a set of predefined morphosyntactic patterns [Evans-1996]. After term candidates are extracted using the patterns, a set of statistical measures, the most simple of them being the frequency, are also used to rank the candidates [Daille-1994].

Of course, real-word systems use both approaches in a higher or lesser extent, and we can consider most of the systems as hybrid [Earl-1970]. With any of these methods we are able to detect a set of term candidates, that is, units with a high chance to be real terms. After the automatic procedure, manual revision must be performed in order to select the real terms from the list of term candidates

Bilingual terminology extraction aims to automatically detect terms in one language and their translation equivalent in another language [Xiong-2016], and they usually work in two phases: the first one for source language term

candidate extraction and the second one for finding the translation candidates. There are some general strategies for the detection of translation equivalents in parallel corpora.

The first strategy is based on the intuition that the translation of a term is likely to be more frequent in the subset of target text segments aligned to source text segments containing the source language term, than in the entire target language text [vanderEijk-1993]. For each source language term, a subcorpus of target text segments aligned to source language segments containing that source language term is created. Then, a terminology extraction is performed in this subcorpus, and the most frequent term candidate is likely to be the translation equivalent of the source term. Other strategies are base on the use of alignment algorithms in the word level [Dagan-1994]. These algorithms are able to relate a word or a group of words in the source text with the corresponding word o group of words in the target text. The third strategy is related to the second one. Some researchers have used phrase tables from statistical machine translation models to find translation equivalents of source terms. For example, [Hjelm-2007] have used GIZA++ [Och-2003] to train translation models to detect term translation equivalents, and have shown that this method outperforms distributional models. [Ideue-2011] uses morphosyntactic patterns in English and Japanese to extract term candidates, and then a phrase table is constructed using Moses [Koehn-2007]. Statistical measures are used to rank the term candidates and extract the highly-ranked ones as bilingual terms.

Others methods are proposed for extracting bilingual terminology from comparable corpora [Fung-1998], but they are out of the scope of this work.

4 Translation Memories, Parallel Corpora and Translation Models

Translation memories and parallel corpora are very valuable resources for translators as *existing translations contain more solutions to more translation problems than any other existing resource* [Isabelle-1992]. Users of CAT tools are automatically creating translation memories as they work. If they perform a good management of their translation memories, in a few months period they will have several thousands of parallel segments with a lot of useful information. If a set of documents and their translations are available, but these translations have been done without the use of a CAT tool, there are several freely available tools performing automatic alignment at the segment level, as for example Hunalign [Varga-2005]. Using this kind of tools large parallel corpora can be compiled in a very fast way. Some parallel corpora are freely available from the Internet and they can be directly used as resource for translation. The most remarkable effort for the compilation of parallel corpora is the Opus Corpus Project.[2] [Tiedemann-2012]

When the size of the available translation memories increases a lot, efficient indexing algorithms are necessary in order to query these resources and get-

[2] http://opus.lingfil.uu.se/

ting a response in a short time. In this work we are using statistical translation models calculated by the Moses toolkit [Koehn-2007]. Although the calculation of the translation models requires a lot of time, once calculated, they are available for as many queries as required. For the automatic detection of the translation equivalent of a given term, we are interested in the phrase translation tables created by Moses. These translation tables provide possible translations for each source n-gram and a set of features in the form of probabilities. These values, along with the values of the target language model, are used by the decoder in order to calculate a final translation with a high probability. Here we can observe a fragment of a phrase translation table:

```
coordinating body ||| el organismo de coordinación ||| 0.222222 ...
coordinating body ||| organismo coordinador de ||| 1 0.072597 ...
coordinating body ||| organismo coordinador ||| 1 0.072597 ...
coordinating body ||| organismo de coordinación ||| 1 0.051453 ...
coordinating body ||| órgano de coordinación ||| 0.5 0.0262583 ...
```

In our system we will use phrase tables to obtain the translation equivalent of a given source term. In the next section more details of how it is done are given.

5 System description

The system we are presenting is developed in two different modules communicated using sockets, to facilitate the use of the system with different CAT tools. One module is independent of the CAT tool and receives the text of the segment being translated at that moment. This module performs ATE and automatic detection of translation equivalents on that segment and shows the results in an interface. The module performs automatic terminology extraction over the received source segment. To rank the term candidates it uses the frequency over all the segments in the translation project and, optionally, the frequency on the reference parallel corpus. As at this moment the source segment is still untranslated, the automatic detection of translation equivalents is performed using the parallel corpus.

The other module is dependent on the CAT tool and detects the change on the segment being translated and sends it to the other module. For the moment this dependent modules is developed only for OmegaT.

To perform all the operations related with terminology extraction and detection of translation equivalents, the system uses TBXTools [Oliver-2015], a Python class that performs several terminology related operations. TBXTools implements both statistical and linguistic terminology extraction methodologies.

– To perform statistical terminology extraction, the n-grams of the segment are calculated. Two kind of filtering with stopwords are performed: all n-grams starting or ending with a stop-word are deleted; a special set of stopwords, called inner stopwords, are used to delete all candidates having

one of these inner stopwords in positions others than the first one and the last one. TBXTools includes a set of stopwords files for several languages that have been automatically created selecting closed-class words from the morphological dictionaries of the Freeling analyzer [Padro-2012]. The list of inner stopwords are created in the same way, but restricting the closed-class words to conjunctions and auxiliary verbs. As these lists are plain text files, they can be modified and extended, and lists for new languages can be easily created.
- To perform linguistic terminology extraction the tool uses an external POS tagger, namely Freeling [Padro-2012], to tag the source segment. A set of terminological POS patterns are used to detect term candidates. TBXTools uses a rich formalism for POS pattern expression allowing searching for word forms, word lemmata and POS tags, and allowing the use of wildcards.

In TBXTools two methods for automatic extraction of translation equivalents are implemented:

- The classical method described in [vanderEijk-1993], where a subcorpus of the target language, having the translation of the source language segments containing the term the translation of which we are searching for. Then the n-grams are calculated and filtered by target language stopwords. The most frequent of these n-grams will be selected as the translation equivalent of the term. In fact, we are performing a statistical automatic term extraction on the subcorpus and the most frequent term candidate is selected as translation equivalent.
- Searches in a translation phrase table from a statistical machine translation model. The first step in this method is the creation of the translation model, in our case using Moses. Then the phrase table is compacted and binarized in order to speed up the retrieval. When we search for a translation equivalent, we consult this phrase table and retrieve all the information ordered by probability. The translator is presented with the n most probable candidates. In the experimental part we have set $n = 5$ so the system provides 5 translation candidates.

The second method is much faster than the first one, provided that we already have the translation model. As the calculation of translation models for very large parallel corpora is slow, the second method is recommended in the case that we want to use such a large parallel corpus and we want to retrieve translation equivalent candidates for a lot of source terms.

We are using this second method in our experiments. The first step is the calculation of the translation model using the parallel corpus. To speed up the search in such big phrase tables, we have created compact phrase tables [Junczys-2012] using the algorithms provided in the Moses distribution. Once created the compact phrase table, we can query it using *queryPhraseTableMin*, a program also distributed with Moses.

Our system allows to add some extra conditions in order to improve the selection of the translation equivalent:

Table 1 Size of the parallel corpora used in the experiments

Corpus	segments	tokens eng	tokens spa
DGT-TM	2,088,196	44,046,368	50,335,833
ECB	116,120	3,107,433	3,488,175
EMEA	1,098,333	12,134,887	13,725,306

- Filtering the candidates with target language stopwords. If the first candidate starts or ends with a stopword, then we reject it and take the next one, and we repeat this verification until the candidate does not start or end with stopwords.
- We consider a minimum and a maximum number of words of the translation candidate. These figures are given as a maximum decrement and maximum increment of the number of words of the source term.

The system can be freely downloaded from SourgeForge.[3]

6 Experimental part

6.1 Experimental settings

For our experiments we've used three parallel corpora, all of them obtained from the Opus Corpus[4] [Tiedemann-2012]:

- For Politics: DGT - A collection of EU translation memories provided by the JRC
- For Finance: ECB - European Central Bank corpus
- For Health: EMEA - European Medicines Agency documents

In table 1 we can observe the size of these corpora.

To create the phrase tables we are using Moses with a language model of order 5 and the default heuristic (*grow-diag-final* to establish word alignments based on the two GIZA++ alignments.

We follow the ideas in [Justeson-1995] to restrict the type and number of terms we take into account in our experiments for each methodology.

- Statistical term extraction: Justeson has analyzed data in dictionaries of technical vocabulary and had found that the majority of technical terms consist of more of than one word. In our experiments we have extracted bigrams and trigrams and we have filtered with a list of 404 stop words.
- Linguistic term extraction: Justeson have also found that the overwhelming majority of terms in probably all domains are noun phrases containing adjectives, nouns, and occasionally prepositions; rarely terms contain verbs, adverbs or conjunctions. Justeson proposes a series of terminological patterns[5]: AN, NN, AAN, ANN, NAN, NNN and NPN (being *of* the most

[3] Bilingual Term Extraction in Real Time: https://sourceforge.net/projects/bteirt/

[4] http://opus.lingfil.uu.se/

[5] Where N: noun; A: adjective and P: preposition

Table 2 Number of words of the English terms in the IATE terminological database

Subject	1	2	3	4	5	>5
04-Politics	9.62	35.05	25.07	13.25	7.62	9.37
24-Finance	7.01	40.58	25.62	13.16	6.27	7.36
2841-Health	30.95	44.48	14.01	5.69	2.57	2.29
Overall	16.30	45.43	20.61	8.88	4.09	4.68

frequent preposition). We have extended this list of patterns and adapted to the tagset used by Freeling for English.[6] Note that the TBXTools formalism for expressing patterns allows the use of wildcards and regular expressions:

```
J.* N.*        J.* J.* N.*
N.* N.*        J.* J.* N.*
VBG N.*        N.* J.* N.*
VBN N.*        N.* N.* N.*
               N.* /of/ N.*
```

In our experiments we are working with terms having 2 or 3 words, as the single word terms are difficult to extract both for statistical and linguistic methods. To support this decision we have performed a statistic of the number of words of the English terms in the IATE (*InterActive Terminology for Europe*[7]) [Johnson-2000] that can be found in table 2. As we can observe, most of the terms have 2 or 3 words (60.12% for politics, 66.2% for finance and 58.49% for health). We can also observe that terms with only one word are residual (less than 10%) for politics and finance, but not for health (30.95%).

In both statistical and linguistic methodology we set a minimum frequency in order to consider a unit as a term candidate. We experiment with two values:

- A minimum frequency of 2 in the document being translated. As the documents we used in the experiments are not very big, this condition might be too restrictive, but using a minimum frequency of 1 would be too permissive and a lot of non-terminological units would be selected as term candidates.
- A minimum value of 25 of the sum of the document frequency and the parallel corpus frequency of the unit.

To find the translation equivalent of each term candidate we have used the translation phrase table procedure and used target language stop-words and set the maximum decrement to 1 and the maximum increment to 2.

[6] The tag set for English follows that of Penn TreeBank for all categories except for punctuation, dates, and numbers, which follow the same general criteria than the other languages in FreeLing.

[7] http://iate.europa.eu/

6.2 Evaluation procedure

The evaluation of automatic terminology extraction is a difficult task. The main difficulty is the impossibility of building a fair gold standard against which the results of the system we wish to evaluate can be compared due to the very low agreement among human evaluators [Vivaldi-2007]. Nevertheless, we have created a test set for each subject. We have downloaded from the Internet 3 documents, one for each of the working subjects and we have randomly selected 200 segments for each document. Then, we have manually selected the 2 and 3-word terms present in the documents (190 terms for Politics, 295 for Finance and 128 for Health) and we have searched for the translation equivalents in external resources. With all these data we can perform an automatic evaluation procedure.

We have performed experiments with the above mentioned three subjects (Politics, Finance and Health), two terminology extraction strategies (statistical and linguistic), two values of the threshold frequency (2 for the document and 25 for the document and parallel corpus).

6.3 Results

In tables from 3 to 5 we can observe the evaluation results for all our experiments. We have evaluated:

- The precision of the term extraction process (ATE P.), counting the number of all the translation candidates and the number of *correct* candidates (that is, the term candidates present in the list of all the terms in the documents).
- The recall of the term extraction process (ATE R.), counting the number of *correct* term candidates and the number of terms in the text.
- The precision of the automatic detection of the equivalent terms in the target language (T.P.) for five positions from the first (T.P.1) to the fifth (T.P.5). The precision for each position includes the previous positions. For example, in T.P.3 we have the precision of the automatic detection of the equivalent terms in the target language for the first, second and third translation candidates.

Observing the values in the tables, we can get some interesting conclusions. Regarding the term extraction process for all the subjects the precision for linguistic methodology is higher than statistical methodology. This higher precision, however, implies a decrease in recall. It is also interesting observing that for all cases, the use of $f_{total} >= 25$ implies a significant increase of recall, as the condition of $f_{proj} >= 2$ is too restrictive, due to the small size of the test documents.

To have an idea of the performance of the automatic term extraction process, we have used TerMine [Frantzi-2000] [8] for the same documents. For Politics, we have obtained a precision of 47.26% and a recall of 38.55%. These

[8] http://nactem.ac.uk

values are better than our values for statistical term extraction, but worse than the linguistic methodology. For Finance, with TerMine a precision of 63.68% and a recall of 48.97% is achieved. These values are in general better than our values, except for the precision of linguistic methodology for $f_{total} >= 25$. In the subject of Health, TerMine is performing better than our system (a precision of 77.91% and a recall o 55.83%). These figures are only indicative, as the compared systems use different methodologies for term extraction. TerMine is using more complex statistical measures for selecting and ranking the term candidates, whereas our systems is using the candidate frequency. In future experiments we will test some statistical measures to improve ATE precision. We must keep in mid, however, than automatic term extraction is only one of the steps in our system. TerMine is not able to detect translation equivalents, so we were not able to compare the performance on this task.

Regarding the automatic detection of the equivalent terms in the target language, most of the correct translations are in the first and second positions, as precision for higher positions does not increase significantly. The best results are obtained for Health, but in our opinion, this is due to the fact that translation equivalents in the Health domain parallel corpus are more consistent than in the other domains.

Table 3 Evaluation results Politics using several strategies

Strategy	ATE P.	ATE R.	T.P.1	T.P.2	T.P.3	T.P.4	T.P.5
Statistic $f_{proj} >= 2$	30.0	46.79	72.55	80.39	80.39	80.39	80.39
Statistic $f_{total} >= 25$	28.37	75.23	76.83	81.71	84.15	84.15	85.37
Linguistic $f_{proj} >= 2$	53.95	37.61	75.61	82.93	82.93	82.93	82.93
Linguistic $f_{total} >= 25$	52.34	61.47	79.1	83.58	86.57	86.57	88.06

Table 4 Evaluation results for for Finance using several strategies

Strategy	ATE P.	ATE R.	T.P.1	T.P.2	T.P.3	T.P.4	T.P.5
Statistic $f_{proj} >= 2$	53.7	10.74	44.83	51.72	51.72	51.72	51.72
Statistic $f_{total} >= 25$	61.61	25.56	66.67	78.26	78.26	79.71	79.71
Linguistic $f_{proj} >= 2$	62.16	8.52	52.17	60.87	60.87	60.87	60.87
Linguistic $f_{total} >= 25$	74.39	22.59	70.49	81.97	81.97	83.61	83.61

Table 5 Evaluation results for for Health using several strategies

Strategy	ATE P.	ATE R.	T.P.1	T.P.2	T.P.3	T.P.4	T.P.5
Statistic $f_{proj} >= 2$	42.35	33.64	63.89	75.00	75.0	75.0	75.00
Statistic $f_{total} >= 25$	37.01	53.27	75.44	85.96	85.96	85.96	87.72
Linguistic $f_{proj} >= 2$	64.44	27.1	65.52	75.86	75.86	75.86	75.86
Linguistic $f_{total} >= 25$	61.11	41.12	72.73	86.36	86.36	86.36	86.36

In table 6 we can observe the mean execution times to process one segment for each methodology. The process of the segment includes the automatic term extraction and the automatic detection of the equivalent terms in the target language. Comparing the methodologies, there are not significant differences

Table 6 Execution time statistics (in seconds)

Subject	Statistical		Linguistic	
	$f_{proj} >= 2$	$f_{total} >= 25$	$f_{proj} >= 2$	$f_{total} >= 25$
Politics	12.38	20.53	10.62	17.36
Finance	4.45	13.91	3.43	12.37
Health	6.13	10.65	5.59	8.70

between them, and we can conclude that the connection to the Freeling server (running in the same computer) is not delaying the process. We need more time to process a segment with $fproj >= 25$ due to the fact that we are obtaining much more term candidates, and we need to find the equivalent in the target language for all of them.

7 Conclusions and future work

In this paper we have presented a system that works along with a CAT Tool and provides automatic terminology extraction and automatic detection of translation equivalents in real time, that is, automatically when the translator goes from one segment to the next one. For automatic terminology extraction the system can use both statistic and linguistic methodologies. The translation equivalents are searched in very large parallel corpora. In order to achieve very fast response times, the system uses compacted and binarized translation phrase tables obtained using the Moses toolkit.

The system can be freely downloaded and holds a free license (GNU GPL). As it is programmed in Python 3, it can be used in most operating systems. The system works along with OmegaT, an open source CAT Tool. In the same web page several translation phrase tables for several language pairs from the Opus Corpus web page can be downloaded. Users can create their own translation phrase tables using Moses.

The system can be very useful for freelance translators and language services providers, as it can save time avoiding performing automatic terminology extraction before starting to translate a project. This task will be done automatically while translating. The system also allows us to calculate translation equivalents for units selected manually by the translator. In this way, the user can find translations for terms missed by the automatic terminology extraction module.

As a future work, we plan to test the system for other language pairs and directions. We also plan to test the system for other domains. To improve the precision of the automatic terminology extraction step, we will test several statistical measures. Once all these improvements will be implemented, we would like to perform a test with real users. The main goal of this test would be to know if the system is really useful for translator and to get their feedback for future improvements.

References

[Arcan-2014] Mihael Arcan, Marco Turchi, Sara Tonelli, and Paul Buitelaar. Enhancing statistical machine translation with bilingual terminology in a cat environment. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2014)*, pages 54–68, 2014.

[Astrakhantsev-2015] N. A. Astrakhantsev, D. G. Fedorenko, and D. Yu. Turdakov. Methods for automatic term recognition in domain-specific text collections: A survey. *Programming and Computer Software*, 41(6):336–349, 2015.

[Bononno-2000] Robert Bononno. Terminology for translators—an implementation of iso 12620. *Meta: Journal des traducteursMeta:/Translators' Journal*, 45(4):646–669, 2000.

[Bourigault-1992] Didier Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 3*, COLING '92, pages 977–981, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[Cabre-2010] Maria Teresa Cabré. Terminology and translation. In Yves Gambier and Luc van Doorslaer, editors, *Handbook of Translation Studies*, pages 356–365. John Benjamins, Amsterdam/Philadelphia, 2010.

[Canovas-2011] Marcos Cánovas and Richard Samson. Open source software in translator training. *Tradumática: traducció i tecnologies de la informació i la comunicació*, 9:46–56, 2011.

[Carretero-2010] Ignacio Carretero. Free software and translation: Omegat, a free software alternative for professional translation. *Teoksessa Boéri, Julie & Carol Maier (toim.). Compromiso Social y Traducción/Interpretación– Translation/Interpreting and Social Activism. ECOS, traductores e intérpretes por la solidaridad, Granada*, pages 146–151, 2010.

[Dagan-1994] Ido Dagan and Ken Church. Termight: Identifying and translating technical terminology. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, ANLC '94, pages 34–40, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[Daille-1994] Béatrice Daille, Éric Gaussier, and Jean-Marc Langé. Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, COLING '94, pages 515–521, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[Earl-1970] Lois L. Earl. Experiments in automatic extracting and indexing. *Information Storage and Retrieval*, 6(4):313 – 330, 1970.

[Eckl-2014] Michael Eckl and Sebastian Haselbeck. Survey of the global translators community 2014. Technical report, LingoIO, 2006.

[Evans-1996] David A. Evans and Chengxiang Zhai. Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 17–24, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[Frantzi-2000] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130, 2000.

[Fung-1998] Pascale Fung. *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas AMTA'98 Langhorne, PA, USA, October 28–31, 1998 Proceedings*, chapter A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-Parallel Corpora, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[Gaussier-2001] Eric Gaussier. General considerations on bilingual terminology extraction. In Didier Bourigault, Christian Jacquemin, and Marie-Claude L'Homme, editors, *Recent Advances in Computational Terminology*, chapter 8, pages 167–183. John Benjamins Publishing Company, Amsterdam/Philadelphia, 2001.

[Heylen-2015] Kris Heylen and Dirk De Hertog. Automatic term extraction. In Hendrik J. Kockaert and Frieda Steurs, editors, *Handbook of Terminology*, volume 1, pages 203–221. John Benjamins Publishing Company, Amsterdam/Philadelphia, 2015.

[Hjelm-2007] Hans Hjelm. Identifying cross language term equivalents using statistical machine translation and distributional association measures. In *Proceedings of NODALIDA*, pages 97–104. Citeseer, 2007.

[Ideue-2011] Masamichi Ideue, Kazuhide Yamamoto, Masao Utiyama, and Eiichiro Sumita. A comparison of unsupervised bilingual term extraction methods using phrase tables. *Proc. MT Summit XIII, Xiamen*, 2011.

[Isabelle-1992] Pierre Isabelle. Bi-textual aids for translators. In *Proc. of the Annual Conference of the UW Center for the New OED and Text Research*, 1992.

[Johnson-2000] Ian Johnson and Alastair MacPhail. Iate-inter-agency terminology exchange: development of a single central terminology database for the institutions and agencies of the european union. In *Workshop on Terminology resources and computation*, 2000.

[Junczys-2012] Marcin Junczys-Dowmunt. Phrasal rank-encoding: Exploiting phrase redundancy and translational relations for phrase table compression. *The Prague Bulletin of Mathematical Linguistics*, 98:63–74, 2012.

[Justeson-1995] John S Justeson and Slava M Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1(01):9–27, 1995.

[Koehn-2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[Och-2003] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[Oliver-2015] Antoni Oliver and Mercè Vàzquez. TBXTools: A free, fast and flexible tool for automatic terminology extraction. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2015)*, pages 473–479, 2015.

[Padro-2012] Lluís Padró and Evgeny Stanilovsky. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May 2012. ELRA.

[Pazienza-2005] Maria Teresa Pazienza, Marco Pennacchiotti, and Fabio Massimo Zanzotto. Terminology extraction: an analysis of linguistic and statistical approaches. In *Knowledge mining*, pages 255–279. Springer, 2005.

[Salton-1975] Gerard Salton, Chung-Shu Yang, and Clement T. Yu. A theory of term importance in automatic text analysis. *Journal of the American society for Information Science*, 26(1):33–44, 1975.

[Tiedemann-2012] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, pages 2214–2218, 2012.

[Varga-2005] Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. Parallel corpora for medium density languages. In *Proceedings of RANLP 2005*, pages 590–596, 2005.

[Vivaldi-2007] Jorge Vivaldi and Horacio Rodríguez. Evaluation of terms and term extraction systems: A practical approach. *Terminology*, 13(2):225–248, 2007.

[Xiong-2016] Deyi Xiong, Fandong Meng, and Qun Liu. Topic-based term translation models for statistical machine translation. *Artificial Intelligence*, 232:54–75, 2016.

[vanderEijk-1993] Pim van der Eijk. Automating the acquisition of bilingual terminology. In *Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics*, EACL '93, pages 113–119, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics.