J Intell Inf Syst (2016) 47:491–514
DOI 10.1007/s10844-015-0382-3

A novel method for dictionary translation

Robert Krajewski[1] · **Henryk Rybinski**[1] · **Marek Kozlowski**[1]

Received: 11 January 2015 / Revised: 18 August 2015 / Accepted: 19 August 2015 /
Published online: 9 September 2015

Abstract The paper addresses the problem of automatic dictionary translation.The proposed method translates a dictionary by means of mining repositories in the source and target languages, without any directly given relationships connecting the two languages. It consists of two stages: (1) translation by lexical similarity, where words are compared graphically, and (2) translation by semantic similarity, where contexts are compared. In the experiments Polish and English version of Wikipedia were used as text corpora. The method and its phases are thoroughly analyzed. The results allow implementing this method in human-in-the-middle systems.

Keywords Dictionary translation · Graphical similarity · Semantic similarity · Multilingual corpus

1 Introduction

Knowledge resources, such as thesauri, taxonomies, and recently ontologies are of high importance in the applications of nowadays information technologies, especially for information retrieval, information extraction, or knowledge discovery methods. Especially with the enormous development of Web the role of knowledge resources has increased

✉ Marek Kozlowski
markozlow@gmail.com

Robert Krajewski
r.krajewski@ii.pw.edu.pl

Henryk Rybinski
h.rybinski@ii.pw.edu.pl

[1] Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland

drastically. However, one of the real barriers in wide use of them is multilinguality of information resources - according to Pimienta et al. (2009), about 65 % of the Internet is a non-English content.

In the context of information retrieval the attempts in solving the multilinguality problems go back to the Salton's works in early 70-ties of the previous century (see e.g. Salton 1970). Since then, several methods of using multilingual dictionaries for improving information retrieval in multilingual text databases have been developed (Hull and Grefenstette 1996; Ballesteros and Croft 1997; Pirkola A 1998). On the other hand, a lot of time and effort has been invested to build and maintain multilingual thesauri and/or flat dictionaries, to be used in enhancing information retrieval in multilingual databases. Many of them (e.g. Eurovoc, GEMET, Agrovoc, INIS) are multilingual and domain oriented. Their translation possibilities are very limited, restricted to the concepts (main descriptors in the thesaurus languages), nevertheless they are extensively used for information retrieval in the international, usually multilingual databases.

Recently, in the context of Semantic Web the problems with multilinguality of the web resources become even more difficult. A vision of a multilingual semantic web has been presented in Gracia et al. (2012). In this vision, a special role is given to the multilingual mappings and linguistic information. In general, it well suits to the idea presented e.g. in Hotho et al. (2003), which consists in seeing ontology as a conceptual layer surrounded by lexical layers. In such approach, each language can be represented by a lexical layer directing from the lexical units in a given language layer to the language-independent concepts and instances belonging to the ontology nucleus (concepts and their instances). What seems to be essential, the lexical layers should cover all the language dependent relationships, including synonymy and polysemy. A more detailed proposal for such an ontology structure is in Wróblewska et al. (2012), and a potential of using such a structure in solving polysemy is presented in Protaziuk et al. (2012).

In order to cope with the multilinguality problems, multilingual components of knowledge resources become of highest importance. Although recently novel methods based on Wikipedia cross-language links have been presented by Sorg and Cimiano (2008a), it seems that specialized multilingual domain-oriented knowledge resources will still play an important role.

In Krajewski et al. (2014) we have presented an approach for translating a lexical layer to a target language. The method, based on mining subject-similar repositories given in the source and target languages, was constructed in such a way that in the first step it discovered a seed dictionary, and then in the second step the seed was used for finding semantically similar terms. Therefore, it was named Seed Based Dictionary Builder (SBDB). Its main feature is that it works without any explicitly predefined relationships between the source and target languages. It is a knowledge-poor approach – it uses merely two text repositories in the source and target languages respectively. To a large extent it is independent of the target and source languages. However, the main drawback of the SBDB method was that the translation was *term-to-term* rather than *meaning-to-term*. As a result, the output dictionary had no references to the meanings of the translated terms.

Protaziuk et al. (2012) proposed incorporating discriminants into the structure of ontology lexical layer, called LEXO. It was shown that such a structure is very useful for the cases of using ontology in text analysis for word sense disambiguation. The representation of term meanings by discriminants is also important in the process of translating dictionaries. In this paper we enhance our method (Krajewski et al. 2014) in such a way that instead

of translating terms the method first discovers meanings of the terms, represented by discriminants. Then it builds the context vectors for the meanings, and finally translates the meanings. We denote the modified method as $SBDB^{+}$. By assigning translations to the meanings rather than the terms we obtain much better precision of the phase of semantic translation. The paper is organized as follows: In Section 2 we discuss related work, then in Section 3 we define basic concepts and formally state the problem. Section 4 presents the algorithm in details. Then we provide experimental results of the method in Section 5. Section 6 concludes the paper.

2 Related work

Following Grefenstette (1993) the methods of text processing can be classified as knowledge-rich or knowledge-poor. The first ones either have a language knowledge embedded within the algorithms processing texts, or are based on using advanced predefined knowledge bases (ontologies, thesauri, semantic dictionaries, etc.). As opposite to knowledge-rich approaches, the latter ones do not use semantic knowledge bases that are difficult and costly to build, and the algorithms used for text processing do not have embedded deep language dependent knowledge. This classification applies also for automatic translation methods, including the research concerning building multilingual dictionaries.

The problems concerning various issues dealing with automatic translation have been widely explored for a long time. In the research, several directions can be distinguished, but the main goal to reduce the human involvement in the translation process, and speed up this process, remains unchanged.

In particular, one of the important research areas, where the translation quality between languages has essential importance is the field of multi-lingual information retrieval (MLIR[1]), where the main objective is to perform search within a multilingual set of documents and collect relevant multilingual documents. Main difficulties resulting from cross-lingual translation ambiguities have been discussed already by Salton (1970).

As a matter of fact, there are many domain-oriented multilingual thesauri (such as Agrovoc, or Eurovoc), which could be used for multilingual information retrieval (Soergel (1997)). They are rich in concepts specific for the domains they are used for, however, there are serious limitations in using them efficiently:

1. The maintenance of multilingual thesauri is quite costly, in addition, they are not flexible enough to follow the domain developments (and linguistic changes in the domain);
2. The lexical granularity is quite sparse and not sufficient for high quality MLIR (usually they contain no more than some 100 000 entries);
3. The semantic relationships between concepts are usually rough, there is no space for polysemy;
4. They require quite a good knowledge from the indexers and end-users to obtain good results.

[1]Following Ballesteros and Croft (1997), and Sorg and Cimiano (2008b), actually two problems are distinguished, namely a more general one, known as multi-lingual information retrieval (MLIR), or more specific one, cross-lingual information retrieval (CLIR).

Hull and Grefenstette (1996) propose using specialized bilingual machine readable dictionaries (MRD). The problem that remains unsolved is to solve disambiguation of such translations. Also Ballesteros and Croft (1997) consider using MRD for translating queries. As an additional mean for improving the retrieval parameters they propose using pre-translation and post-translation feedbacks.

A representative method within CLIR is the approach named *Cross Lingual Latent Semantic Analysis* (LSA) described in Dumais et al. (1997). This method consists in performing SVD on vectors built with term frequencies in documents and its translated versions. Between the vectors built in a latent space, a similarity can be computed. Especially, new vector (representing a document or a query) could be added to the latent space, and its similarity to other vectors in the space can be measured.

An interesting approach, called CL-ESA (*Cross Lingual - Explicit Semantic Analysis*), has been presented in Sorg and Cimiano (2008a). The approach is an extension of Explicit Semantic Analysis (ESA), the idea presented in Gabrilovich and Markovitch (2007). Both, ESA and CL-ESA can be definitely classified as knowledge- rich, as they are based on extensive use of Wikipedia as a source of a text repository (articles), but also as a provider of semantic knowledge (ESA) and a structured cross-lingual relationships between Wikipedia articles. Given a document, CL-ESA uses a document-aligned cross-lingual reference collection (CL-ESA) in Wikipedia to represent the document as a language-independent concept vector (usually expressed by $tf\text{-}idf$). In CL-ESA, the relatedness of two documents in different languages is assessed by the cosine similarity between the corresponding vector representations. Cimiano et al. (2009) have shown that in the field of CLIR CL-ESA approach outperforms latent concept models in terms of result quality, but also by means of performance. A good example of applying knowledge-rich approach for building BabelNet is presented by Navigli and Ponzetto (2012). BabelNet is a multilingual semantic network and is obtained from the automatic integration of WordNet, Wikipedia, Wiktionary and WikiData.

Other semantic support in translating the dictionaries can be taken from existing ontologies or thesauri. A method combining statistical and semantic approaches for translating thesauri and ontologies is presented in McCrae et al. (2011).

Unfortunately, these methods have limitations. Namely, they cannot be used when the semantic resources (Wikipedia, thesauri or ontologies) are insufficient. For Wikipedia, even for the languages well represented, it may turn out that some domain-specific parts are missing or are under-represented. On the other hand, many multilingual thesauri (such as Agrovoc, or Eurovoc) are rich from the point of view of the domain semantic, and the semantic contents can be a good starting point, especially while building a lexical layer for a domain ontology, however the lexical volume of the thesauri is quite limited (usually they contain no more than some 50 000 entries, and they ignore polysemy).

Among other knowledge-rich approaches, there are also syntax based methods for exploiting grammar structure of sentences (Yamada and Knight (2001)). This kind of approach, in opposite to the methods presented above, belong to the family of deep analysis methods. They are characterized by employing a knowledge about the syntactic or even semantic structure of given texts. The main problem with these approaches is that they are strongly language dependent and cannot be easily adopted to another pair of languages.

Another translation problem refers to the task of translating ontologies or thesauri. In this case where the text data are semi-structured, a support in translating such structures can be taken from semantic relationship contained in the structure. A method combining statistical and semantic approaches for translating thesauri and ontologies is presented in

McCrae et al. (2011). Recently, in the context of linked data the problem of language-independent reasoning in Semantic Web was analyzed in Gracia et al. (2012). The authors discuss various layouts of ontologies and their lexical surroundings. Also in this case the lexicons for translating between the languages are shown as indispensable.

In the area of machine translation there are numerous knowledge-poor algorithms, which are based on statistical methods, and could be used for translating word-to-word or word-to-phrase between two languages. One of the first systems of this kind was IBM Model 1, presented by Brown et al. (1990). It was based on the EM algorithm and provided word-to-word translation. Then the approach has been extended by Koehn et al. (2003) for word-to-phrase translating. The idea has been further developed by Vogel et al. (1996) and Deng and Byrne (2008) for word-to-phrase translation, and with the use of the Hidden Markov Models. Based on features of IBM Model 4, Deng and Byrne (2008) improve essentially the translation quality by using the HMM models. The translation precision of the methods can reach a very high level. Unfortunately, the main problem with the approach is that it has to use large parallel repositories, whereas for most language pairs, parallel texts are hard to find. Even for texts derived from official multilingual speeches (e.g. from EU parliament) may not be appropriate for a particular subject domain.

Especially in the context of domain ontologies for such domains that witness dynamic development there is a high interest for lexicon translation methods, which are knowledge-poor and do not rely on parallel repositories. The idea of lexicon translation with nonparallel corpora has been presented in Rapp (1995). The presented method is based on the observation that if the words a and b collocate often in one language then their translations should collocate in the repository in the target language too. Additionally, for large corpora the frequencies of the collocations should be similar. However conceptually correct, this solution turns to be computationally very expensive. The problem of how to obtain a starting seed lexicon is not considered in the paper.

An interesting solution for obtaining the seed lexicon has been proposed by Koehn and Knight (2002). Namely the authors postulate building the seed translation dictionary from the source and target repositories, based on the existence of some words in both languages in the same form, or similar in terms of spelling of the corresponding source and target words.

Our approach for building bilingual dictionaries has been influenced by Rapp (1999) and Koehn and Knight (2002). Also in our approach we build a seed dictionary within the first phase of building bilingual dictionary, but opposite to Koehn and Knight (2002), we show that knowledge-poor data mining methods can be used successfully even for the languages belonging to different families (English and Polish). Having built the seed, we also build vectors for the source terms, and translate them with the seed dictionary, but the way the vectors are constructed is different. The main difference consists in finding various meanings of the terms to be translated, and then in finding the translations for the meanings, rather then terms. In Sections 3 and 4 we present our approach in more detail.

3 Basic concepts and definitions

In this Section we state formally the problem of building a bi-lingual dictionary. Given a repository R in a language L we can extract a dictionary D, composed of *simple terms* (one-word terms) or *compound terms* (multi-word terms). As in (Rybinski et al. 2008), any set $X \subseteq D$ of terms will be called *termset*. Significance of a termset X, called *support* and denoted by $sup(X)$, is expressed by the number of paragraphs of R, in which X appears.

Termset X is called $frequent$ if it occurs in more than ϵ paragraphs in R, where $\epsilon > 0$ is a user-defined support threshold. We say that X is a $context$ of term t if $sup(tX) > 0$ in R, and it is *frequent context* of t if $sup(tX) > \epsilon$ in R. Distinct meanings of terms are indicated by distinct contexts in which they appear frequently. This assumption is based on the distributional hypothesis (Harris (1981)), where the underlying idea is that "a word is characterized by the company it keeps". The rule is very intuitive, the problem is, however, how the notion of a context is defined. To distinguish various meanings of a term t we introduce the concept of discriminants. Discriminants play a role of such contexts that define various "meanings" of the lexical unit t. We also distinguish a set of meanings $\mathfrak{M}$, and treat discriminant as a representation of a meaning $m \in \mathfrak{M}$. Below we present an idea of how the meanings of t can be represented. Given term t and the termsets X and Y, we define relative support as:

$$relSup(t, X, Y) = \min\{sup(tXY)/sup(tX), sup(tXY)/sup(tY)\} \tag{1}$$

The relative support measures how "distinct" in terms of support are collocations tX and tY. Provided both tX and tY are frequent, if X and Y belong to different domains the support of the combination of t, X and Y together is usually much lower. For example, having the frequent termsets *apple*, *fruit* and *apple*, *motherboard*, we can expect that the support of *apple*, *fruit*, *motherboard* will be much lower from at least one of the original termsets.

Definition 1 The termset X is called a δ-discriminant for t if for a given $\delta > 0$ the following conditions are satisfied:

1. the termset tX is frequent
2. there exists a termset Y, for which tY is also frequent in R
3. $relSup(t, X, Y) \leq \delta$.

We say that for the term t the termsets X and Y are δ-discriminants against each other.

We say that X discriminates a meaning of t against Y, or X and Y are against each other for term t. We are interested in minimal discriminants (i.e. such discriminants that cannot be reduced). Term t may have many discriminants. We denote them by $Disc(t)$. In other words, for given t we have $d_1 \in Disc(t)$ iff there is $d_2 \in Disc(t)$ and d_1 is discriminant against d_2 for t. Not every pair of discriminants from $Disc(t)$ are against each other. For example, for the term $apple$ the termset *processing unit* is discriminant against *juice*, but it is not discriminant against *motherboard* (which is another discriminant of the term *apple*, and is against *juice*). Bearing this in mind, we can cluster the discriminants from $Disc(t)$ in such a way that in each cluster the discriminants are not against each other. For $d_1, d_2 \in Disc(t)$, such that d_1 is against d_2, we write $d_1 \# d_2$. So, we can cluster all the discriminants from $Disc(t)$ in the following way:

$$C_t = \{C_t^1, \ldots, C_t^k\}, \text{ where}$$
$$C_t^i = \{d : d \in Disc(t) \wedge \forall_{d' \in C_t^i} \neg(d \# d') \wedge \forall_{j \neq i, d' \in C_t^j} d \# d'\}, i \in \{1, \ldots, k\} \tag{2}$$

For a term t we have thus k_t clusters. The discriminants within each cluster are compatible, whereas any two discriminants selected from any two various clusters from C_t are against each other. We assign each $C_t^i \in C_t$ a meaning from $\mathfrak{M}$.

Table 1 The supports of termsets combining *apple* with terms from X

$sup(\bullet, \bullet)$	Computer	Company	Fruit	iphone	Tree
Apple	5510	6021	2719	1251	3561

Let us denote by $\mathbf{C}$ the set of all clusters of discriminants for all the terms in D:

$$\mathbf{C} = \bigcup_{t \in D} C_t$$

Let us illustrate the introduced notions by the following example

Example 1 Let us consider a word *apple* with its two meanings — a fruit hanging on a tree or the well known tech company. In order to illustrate how the discriminants can be found for these meanings we use the English version of Wikipedia.[2] Wikipedia includes 22276 articles containing the word *apple*. For the purpose of this example we consider the set $X = \{computer, company, fruit, iphone, tree\}$, potentially containing discriminants of the two meanings. In Table 1 the supports of termsets combining *apple* with terms from X are presented.

Now, in order to find out discriminants we have to measure $relSup$ for $\{apple, x, y\}$ for $x, y \in X$. Table 2 shows the supports of triples $\{apple, x, y\}$, so that we can calculate the needed $relSup(apple, x, y)$ measures.

With the values of $sup(apple, iphone)$, $sup(apple, fruit)$ from Table 1, and $sup(apple, iphone, fruit)$ from Table 2 we obtain $relSup(apple, iphone, fruit) = 0,005149$. For the threshold $\delta = 0,01$ and for $t = apple$ the terms $fruit$ and $company$ are discriminants against each other, and the terms $iphone$ and $fruit$ are discriminants against each other. Let us now verify if the term $company$ is discriminating any of those two meanings into other meanings: from Tables 1 and 2 we can evaluate $relSup(apple, iphone, company) = 0,098$ and $relSup(apple, fruit, company) = 0,105$, which means that for $t = apple$ the term $company$ is nor discriminant against $fruit$ nor against $company$. In other words, it can appear in the contexts of both meanings – apple as a fruit and apple as a company name.

We can now define semantics of the terms from dictionary D by a mapping that assigns a meaning to each pair (t, C_t^i), i.e. $Sem: D \times \mathbf{C} \to \mathfrak{M}$, so that for any $t \in D$ and $C_t^i \in C_t$ there is a meaning $m \in \mathfrak{M}$. Given $Sem(t, C) = m, t \in D, C \in \mathbf{C}$, we say that m is the meaning of t in the context of C. We do not impose any restrictions on the mapping Sem for assigning the same meaning to two various arguments, so we can have two terms p and q, $p \neq q$, such that for certain $C_p^i, C_q^j \in \mathbf{C}$ we have $Sem(p, C_p^i) = Sem(q, C_q^j)$, which means that p and q are synonymic for some contexts, though do not have to be synonymic for other contexts.

Now we introduce our primary goal, which can be expressed as follows: We have given two repositories R_S and R_T of texts in the source and target languages L_S and L_T , respectively. The repositories determine flat dictionaries D_S and D_T , specific for L_S and L_T , and resulting from the (possibly independent) repositories R_S and R_T . The dictionaries may contain single words, as well as, compound terms (proper names, compound terms).

[2]In our experiment Wikipedia is used solely as a repository. For this experiment any other repository could be used. For example in Nykiel and Rybinski (2008) the Google resources were used instead.

Table 2 Supports of $\{apple, x, y\}$

$sup(apple, \bullet, \bullet)$	Computer	Company	Fruit	iphone
tree	428	847	1094	66
iphone	547	588	14	
Fruit	151	634		
Company	2185			

Problem Given D_S and D_T find a translation function

$$\Gamma\colon D_S \times \mathbf{C} \to 2^{D_T},$$

such that for $p \in D_S$, and $C_p^i \in C_p$ we have $\Gamma(p, C_p^i) = \{t_1, \ldots, t_k\}$, where $t_i \in D_T$ is a translation of the meaning of p in the context C_p^i (i.e. $Sem(p, C_p^i) = m$), to the target language L_T. The set $\{t_1, \ldots, t_k\}$ is a set of synonyms in L_T, corresponding to the meaning m.

Below, we will propose a solution that supports building the translation function Γ.

4 The $SBDB^+$ algorithm

4.1 Brief description of the algorithm

As mentioned in Section 2, our approach has been influenced by Rapp (1999), and Koehn and Knight (2002). However our goal differs from the one presented in Koehn and Knight (2002). As specified in Section 3 we tend to find translations for the meanings expressed in the source language, rather than the words. What is important, the proposed method uses knowledge-poor text mining algorithms. It consists of four phases:

1. First, for the repositories R_S and R_T the monolingual dictionaries D_S and D_T are extracted.
2. Given D_S and D_T we build a bilingual seed dictionary.
3. In the third phase, in the repository D_S for each source term t we mine for the discriminant sets C_t
4. The fourth phase is devoted to building „context vectors"; for the source language we use the R_S repository and build them for the pairs (t, C_t^i), whereas for the repository R_T the context vectors are built for the terms $s \in D_T$; then, with the use of the seed dictionary built in Phase 2 we look for the most similar translate candidates

A conceptual diagram of the method is presented in Fig. 1. The first phase of extracting dictionary is a standard procedure, where we reject stop words and select only specific parts of speech.

The second phase is devoted to generate a seed dictionary. The idea is that words from two repositories – source and target, denoted by D_S and D_T respectively – are compared, and the similarity translation function (bijection) $\gamma : D'_S \to D'_T$, $D'_S \subseteq D_S$, and $D'_T \subseteq D_T$[3] is built by using an edit distance measure. The phase is similar to

[3]The domain and codomain of γ are denoted by $Dom(\gamma)$ and $CoDom(\gamma)$ respectively.

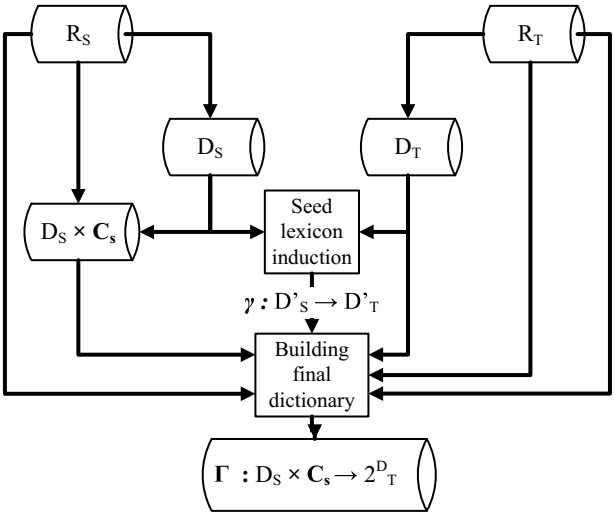

Fig. 1 Conceptual diagram of the method

the one presented in Koehn and Knight (2002), but opposite to it, no translation rules have to be predefined. Instead, during the comparison process the algorithm mines for the transformation rules that are specific for the source and target languages, S and T respectively. The rules are dynamically added to the set of rules, which are then reused in the continued comparison process. It means that no extra language-dependent knowledge about the rules for transcription from the source language to the target one is needed. We call this phase *syntactic translation*, as the pairs are identified by syntactic similarities.

The third phase is crucial for the quality context vectors that are built in Step 4. It is used to identify the discriminants for each $t \in D_S$, so that a set of meanings $\mathfrak{M}$ is built for the terms in the source language.

In Phase 4, the results of the two previous phases (2 and 3) are used for building the translation function Γ, as defined in Section 3. In order to reach this goal, for every pair (s, C_s^i), $s \in D_S$, $C_s^i \in C_s$, a context vector is built, and then the semantically close context vectors from D_T are identified. The semantic relatedness can be identified as the seed dictionary can be used. As a result of this phase for the pairs (s, C_s^i) the k best candidates are provided, so that the final work can be easily performed manually. This phase is called *semantic translation*, as it mines contexts in the source and target languages and looks for the semantic similarities.

In the consecutive subsections the particular phases of the algorithm are presented in detail.

4.2 Lexical similarity

The idea of using lexical similarity is based on the assumption that some words exist in a similar form in the source and target languages. Actually, the similarities between languages are subject of intensive research by the linguists. McMahon (1994) discusses how the languages belonging to the same family diverge, whereas Kranich et al.

(2011) give an interesting example how the languages belonging to different families influence each other. It seems therefore to be straightforward to apply the similarities for building seed dictionaries not only for the languages belonging to the same family.

Linguists indicate various reasons of similarities between the dictionaries of the different languages. For the languages belonging to the same group, to large extent the common roots in the past decide on the similarities. Additionally, an influence of one language to another one may result from various factors, such as, e.g. technology transfer, cultural influence, etc. This may also refer to the languages belonging to different groups, like e.g. German and Polish.

The words having similar form (and meaning) are named by linguists cognates, and it turns out that they appear quite often across the modern languages. The problem with cognates is that they are in slightly different forms (due to the linguistic rules in the source and target languages). To perform transcription from one form to another Koehn and Knight (2002) used linguistic knowledge and defined a set of rules. Instead, we propose to mine out the rules automatically while searching for cognates, so that no specific knowledge is needed.

To measure similarity between words an edit distance computation algorithm can be employed. We will use a slightly modified Damerau-Levenshtein edit distance measure. Usually the distance measure between the terms p and q is defined as follows: $d(p,q) = n/max\{l(p), l(q)\}$, where n is the number of changes (in characters), which have to be performed to achieve q from p, and $l(t)$ is the length (in characters) of term t. In our case we reduce the measure by the number of frequent rules that are already applicable to the term translation (see Definition 2).

The proposed algorithm differs from the ones known from the literature that along with the process of looking for similar words it additionally mines the rules. So, if a given type of differences between similar words appears often enough, the algorithm adds new translation rules to the set of rules and continues building the dictionary with the use of the mined rules.

The method extracting seed lexicon is designed as an iterative process of building incrementally a bijection function $\gamma : D'_S \rightarrow D'_T$, $D'_S \subseteq D_S$, and $D'_T \subseteq D_T$. Schematically it is shown in the form of pseudocode for Algorithm 1. Each iteration consists of two steps:

1. Given D_S and D_T find two sets:
   (a) $\Delta\gamma$ containing all pairs $(s,t) \in D_S \times D_T$, such that t can be reached from s by applying the translation rules from the set $\mathcal{R}$ (empty at the beginning);
   (b) σ containing all candidate pairs, i.e. such pairs that the distance between s and t is minimal and not higher than a given threshold.

   Add $\Delta\gamma$ to γ and remove the matching words from D_S and D_T, so that they will not participate in the next iterations;
2. For every pair $(s,t) \in \sigma$ find all differences, and for each difference build a candidate of translation rule.
   (a) If the rule exist in the set of rules $\mathcal{R}$, increase the rule support, otherwise, check if the rule exist in the set of candidate rules $\mathcal{R}_c$ - if so, increase the support, if not, add the rule to $\mathcal{R}_c$.
   (b) When passed all the pairs from σ, for every candidate translation rule $r \in \mathcal{R}_c$ such that $sup(r) > \delta$, add the rule to the set of frequent rules $\mathcal{R}$.

3. If no new rule is added to the set of rules $\mathcal{R}$ then terminate, otherwise go to 1.

First, let us define the notion of transformation rule r. The rule has the form $x \rightarrow y, x \in \Sigma_S^*$, and $y \in \Sigma_T^*$, where Σ_S, and Σ_T are the alphabets of the languages S and T[4].

Algorithm 1 Computation of γ – lexical similarity dictionary

```
1: function LEXICAL-SIM(D_S, D_T, μ)
2:   γ ← ∅
3:   R ← ∅
4:
5:   R_c ← ∅
6:   Δγ ← TRANSLATE(D_S, D_T, R, 0)
7:   While Δγ ≠ ∅ do
8:     σ ← TRANSLATE(D_S, D_T, R, μ)
9:     γ ← γ ∪ Δγ
10:    for (s, t) ∈ Δγ do
11:      D_S = D_S − s
12:      D_T = D_T − t
13:    end for
14:    for (s, t) ∈ σ do
15:      for r ∈ RULES(s, t) do
16:        if r ∈ R do
17:          sup(r, R) = sup(r, R) + 1
18:        else if r ∈ R_c then
19:          sup(r, R_c) = sup(r, R_c) + 1
20:        else
21:          R_c = R_c ∪ {r}; sup(r, R) = 1
22:        end if
23:      end for
24:    end for
25:    for r ∈ R_c do
26:      if sup(r, R_c) > δ then
27:        R = R ∪ {r}
28:        R_c = R_c − {r}
29:      end if
30:    end for
31:    Δγ ← TRANSLATE(D_S, D_T, R, 0)
32:  end while
33:  return γ
34: end function
```

[4]We presume that a large part of the alphabets of the source and target languages is shared by both languages, which is the case when e.g. both languages are based on the Latin alphabet.

Given a set of rules ρ, and the terms p and q we say that p can be transformed to q with ρ iff every rule $x \rightarrow y$ from ρ is applied at least once in substituting the substring x of p into y, and the final term obtained is q. For this we write $p \xrightarrow{\rho} q$, and say that ρ is the edit difference set of rules for the pair (p, q). In order to indicate that ρ refers to the pair (p, q), whenever needed we will write ρ_{pq} For example we say that

1. *orthography* can be transformed to *ortografia* with the set of rules $\rho = \{th \rightarrow t, ph \rightarrow f\}$
2. *comic* can be transformed to *komik* with the set of rules $\rho = \{c \rightarrow k\}$

We call the rule r frequent if in the process of translating from D_S to D_T, it is applied more than δ times, where δ is a user-predefined threshold, i.e. $sup(r) > \delta$. At i-th iteration of computing the translation function γ, the set $\mathcal{R}$ of frequent transformation rules is available. Now, let us present in detail the way how the procedure TRANSLATE works. The procedure uses a modified edit distance measure between terms t and s. Actually, we define the distance measure as a function of terms t, s, and a set of rules $\mathcal{R}$.

Definition 2 Given a pair of terms (s, t) and the edit difference set of rules ρ for this pair, we define the distance between s and t as $d_{\mathcal{R}}(s, t) = d(s, t) - ||\rho \cap \mathcal{R}||$, where $d(s, t)$ is the standard Damerau-Levenshtein measure, and $||\rho \cap \mathcal{R}|| = |\rho \cap \mathcal{R}|/max\{l(t), l(s)\}$.

The pseudocode of the procedure TRANSLATE is presented as Algorithm 2. The procedure can work in two modes, namely as a procedure discovering the translations, and as the procedure discovering the translation candidates. In the first case (the parameter $\mu = 0$), given $s \in D_S$ and $\mathcal{R}$, the procedure TRANSLATE looks for $t \in D_T$ such that $d_{\mathcal{R}}(s, t) = 0$, i.e. $\rho_{st} \subseteq \mathcal{R}$. If there are more possible translations of s than one, the procedure selects t_0, which maximizes the total support of ρ in $\mathcal{R}$, i.e:

$$\forall t \in D_T \sum_{r \in \rho_{st}} sup(r) \leq \sum_{r \in \rho_{st_0}} sup(r) \tag{3}$$

As we can see, for each $s \in D_S$ we look for $t_0 \in D_T$ such that the distance $d_R(s, t_0) = 0$, and the condition (3) is satisfied for t_0. If t_0 is found, the pair (s, t_0) is added to the result function γ and t and s_0 are removed from D_S and D_T respectively, which guarantees that γ is a bijection.

In the second case ($\mu > 0$), for each $s \in D_S$ the procedure TRANSLATE finds out such candidates $t \in D_T$ that the distance between s and t is less than μ. The candidates may become translations if at a given iteration the corresponding candidate translation rules from $\mathcal{R}_c$ become frequent.

The process of extracting rules by comparing source and target terms *copy* and *kopia* is visualized in Fig. 2. There are two contiguous sequences generating candidate rules, namely (c, k) and (y, ia), so that we have $\rho_{copy,kopia} = \{c \rightarrow k, y \rightarrow ia\}$. In the case when $\rho_{copy,kopia} \subseteq \mathcal{R}$ the pair $(copy, kopia)$ is added to γ. Otherwise, if $d_{\mathcal{R}}(copy, kopia) \leq \mu$ the pair $(copy, kopia)$ is added to the set σ of candidate pairs, and the support of rules from $\rho_{copy,kopia}$ is recalculated.

Algorithm 2 Translation function

```
1: function TRANSLATE(D_S, D_T, R, μ)
2:   m ← μ
3:   Δγ ← ∅
4:   for s ∈ D_S do
5:     maxsup = 0
6:     p ← null
7:     for t ∈ D_T
8:       if μ ≥ d_R(s, t) then
9:         if sumsup(s, t) ≥ maxsup and d_R(s, t) = m then
10:          m ← d_R(s, t)
11:          p ← (s, t)
12:          maxsup = sumsup(s, t)
13:        else if d_R(s, t) < m then
14:          m ← d_R(s, t)
15:          p ← (s, t)
16:          maxsup = 0
17:        end if
18:      end if
19:    end for
20:    if notnull(p) then
21:      (a, b) ← p
22:      D_S = D_S − {a}
23:      D_T = D_T − {b}
24:      Δγ ← (a, b)
25:    end if
26:  end for
27:  return Δγ
28: end function
```

As a result of the function LEXICAL-SIM we receive a seed dictionary in a form of the translation function γ. The details concerning the quality of the translation will be discussed in Section 5. In the next subsection we will present how with a seed dictionary the semantic translation can be performed.

4.3 Semantic similarity

Within this phase of the method, a semantic similarity analysis is performed for the terms to be translated, so that the previously created seed dictionary is extended. A general idea, originated by Rapp (1999), is that in similar target and source repositories the statistical distribution of collocations for the corresponding terms, source and its translation, should be very similar to each other. Therefore the algorithm is based on contexts found out for each term extracted from the source and destination repositories, R_S and R_T respectively. In order to find semantically similar terms, one should construct *context vectors* for the terms in D_S and D_T, reflecting relationships between the terms and the terms from the seed dictionary γ, and then look for the vectors having similar distribution characteristics. Hence, given contexts (termsets) for all the terms $x \in D_S$ we limit them only to those termsets that

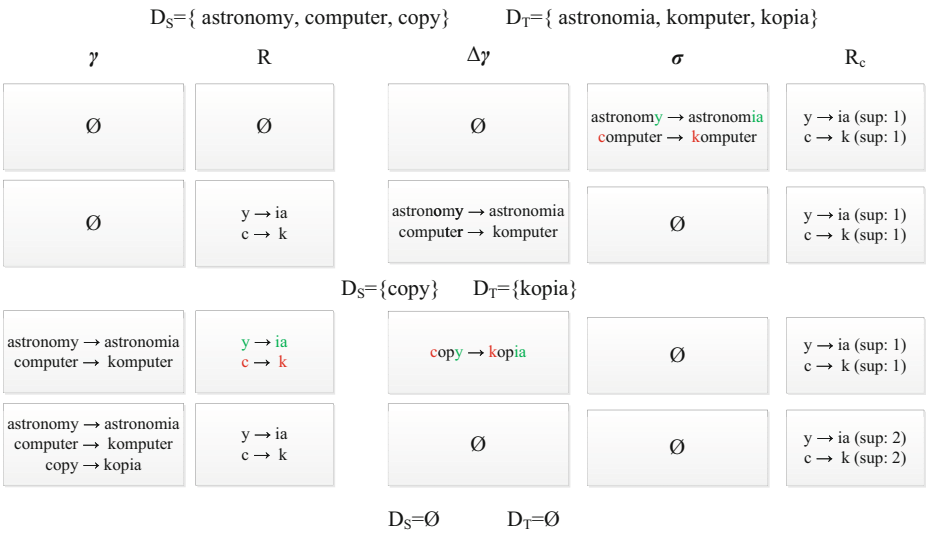

Fig. 2 An example of $lexical - sim$ algorithm for $S = \{\text{computer, astronomy, copy}\}$ and $T = \{\text{komputer, astronomia, kopia}\}$

contain $t \in Dom(\gamma)$, and replace t by $\gamma(t)$ in these termsets. Then limiting the termsets only to the terms from $Dom(\gamma)$ we can build the context vectors, where the i–th position is the frequency of t_i. In a similar way we proceed with the terms $y \in D_T$, limiting the context vectors to the terms $t' \in CoDom(\gamma)$. Now having vectors for the source and target languages, for each x we can search for the closest y by means of the $cosine$ pseudometrics.

However, the main problem with this approach is that the context vector for a term from the source dictionary may cumulate dimensions specific for various meanings of this term. To illustrate it, let us consider the word $vessel$. The term can be used as a ship, blood vessel, or a dish. In each case the accompanying context terms are totally different, so if we do not split the meanings, all the three characteristics will mix up in one context vectors. Depending on the repository, some of the meanings represented in the context vector can be dominating. On the other hand in the target language, the three meanings may have different word representations, so we can expect that separate vectors, free of the other meanings components, will be created. The relatedness between the vector of $vessel$ and its three corresponding vectors may become too difficult to discover, if possible at all.

The importance of this problem becomes much clearer when we look at the statistics for the natural languages. As stated in Miller et al. (1994), about 73 % of words in common English are polysemous, the average number of senses per word for all words found in English texts is about 6,5 (Mihalcea and Moldovan 2001). To this end, in our approach instead of building vectors for terms $x \in D_S$, we build them for the pairs (x, m), $m \in \mathfrak{M}$. As a result, we obtain contexts vectors specific for the meanings.

More formally, this part of the algorithm, responsible for semantic translation, can be sketched by means of the following steps:

1. Given the dictionary D_S, we mine for the set of meanings $\mathfrak{M}$, i.e. to each $t \in D_S$ we assign a set of meanings $\{m_1, ..., m_{k_t}\}$, $m_i \in \mathfrak{M}$. If no discriminants are identified for t, it represents one meaning;

2. For each pair (t, m), $t \in D_S$, $m \in \mathfrak{M}$ the termsets are identified from the repository R_S. The termsets are built based on paragraphs containing t or $2w$ words windows around t. By W_{tm} we denote the set of all termsets built for (t, m);
3. For each pair (t, m) we build W_{tm}^{γ} – we namely filter out of W_{tm} only those termsets that contain terms from the seed γ, as calculated in Phase 1 (see p. 4.2);
4. Now, based on the sets W_{tm}^{γ}, for each pair (t, m) we aggregate the semantic context vectors, building the vector space;
5. In order to reduce noise in the vectors we leave the top l positive dimensions, setting to zero the remaining ones;
6. In a similar way we build vector space for the target repository, except that it is constructed for terms instead of the pairs (t, m), so actually no meaning set is detected;
7. For each vector from V_S we look for the closest k vectors v from V_T.

Below we discuss all the steps in detail.

4.3.1 Finding meanings

As mentioned above, the first step of this phase is crucial for the quality of the semantic translation. Comparing this method to the original seed based methods (Rapp (1999), Koehn and Knight (2002), as well as Krajewski et al. (2014)), having split the meanings of each t, and building the context vectors for (t, m) instead of t, we obtain context vectors much more specific for the particular meanings, drastically reducing the noise usually cumulated in the vectors.

In order to perform Step 1 we apply the method called SenseSearcher, in short SnS ((Kozlowski 2014), Kozlowski and Rybinski (2014)). It is a knowledge-poor word sense induction algorithm based on closed frequent set. For the reasons listed below it is especially well suited for our purposes:

– it is able to find infrequent, dominated senses;
– the proposed method creates structure of senses, where coarse-grained senses contain related sub-senses (fine-grained senses), rather than a flat list of concepts:
– the number of discovered senses does not have to be predefined by the user; it is determined solely by the content of the corpus;
– as an output it provides the meaning discriminants, which can be directly used for building the representations of the pairs (t, m) for the consecutive steps of the translation;
– its quality outperforms the existing state-of-the-art methods in most cases;

Below, we describe the algorithm in more detail. The pseudocode of the whole algorithm is presented as Algorithm 3. It consists of five phases:

– Phase I is devoted to building an inverted index for the corpus. Each document is split into paragraphs (the paragraphs are used to build frequent termsets);
– in Phase II, for each term a query with the given term is performed on the index, so that the paragraphs containing the term are found. The termsets are converted into the context representations. The algorithm forms as many contexts as many paragraphs are being analyzed.
– Phase III is devoted to generating contextual patterns from the contexts generated in the previous phase. The contextual patterns are closed frequent termsets identified in the context space;

Algorithm 3 SnS algorithm - the Word Sense Induction method

Input: $Corp$ as a textual corpus, t as a queried term and ϵ as a minimal support, σ and γ as similarity thresholds

Output: $S(Corp, t, \epsilon, \sigma, \gamma)$ as a set of senses

Phase I - `for each document` $d \in Corp$ `paragraphs are extracted and added to full-text search index` $L(Corp)$

Phase II - `the full-text search index` $L(Corp)$ `is queried for a term` t `and retrieves` $\mathcal{P}(t)$ `that are converted into` $\mathcal{C}$

$\mathcal{C} = \{\}$

for paragraph $p \in \mathcal{P}(t)$ **do**

$c = buildContext(p)$ ▷ convert p into a context c, persisting only proper names, named entities and nouns

$\mathcal{C} = \mathcal{C} \cup c$

end for

Phase III - `discovering contextual patterns` $\mathcal{CP}$ `from contexts` $\mathcal{C}$, `and sorting them by support descending`

$\mathcal{CP} = contextualPatternsMining(\mathcal{C}, \epsilon)$

Phase IV - `building sense frames` $\mathcal{SF}$ `from` $\mathcal{CP}$

$LC = \{\}, LT = \{\}, \mathcal{SF} = <>$

for contextual pattern $cp \in \mathcal{CP}$ **do**

if $cp \notin LC$ and $cp \cap LT = \emptyset$ **then**

$mp = cp$ ▷ cp becomes main contextual pattern mp

$sf = SenseFrame(mainContextPattern = mp, sTree = \emptyset)$

$scp = findSubContextPatterns(mp)$ ▷ find supersets of mp among $\mathcal{CP}$

$LC = LC \cup mp \cup scp; LT = terms(LC)$ ▷ expand locked contexts and locked terms

$buildSenseFrame(sf, scp)$ ▷ organize scp in a tree hierarchy and assign to $sf.sTree$

$\mathcal{SF}.append(sf)$

end if

end for

Phase V - `clustering sense frames` $\mathcal{SF}$ `in order to find tight senses`

$clusters = senseFramesClustering(\mathcal{SF}, \sigma, \gamma)$ ▷ clusters of similar sense frames

$labeledSenses = labelSenses(clusters)$ ▷ clusters are labeled by the most common noun terms among grouped sense frames

$S(Corp, t, \epsilon, \sigma, \gamma) = labeledSenses$ ▷ labeled sense contains descriptive sense label, and set of incorporated sense frames

return $S(Corp, t, \epsilon, \sigma, \gamma)$

– Phase IV is devoted to forming contextual patterns into sense frames, building multi-hierarchical structures corresponding to senses. In some exceptional cases few sense frames may refer to the same sense, it is connected with the size of input document

corpus (lack of representativeness and high synonymy similarity against descriptiveness of terms);
– in Phase V, sense frames are clustered. The clusters of sense frames are called senses. Optionally senses can be labeled with some descriptive terms.

Having completed the last phase of the algorithm, each group of similar sense frames can be treated as a coherent meaning, and the input paragraphs are partially assigned to matching senses. Given a term t, we can now identify its discriminants C_t, and group them into clusters $C_t = \{C_t^1, \ldots, C_t^k\}$, according to the condition (2), i.e. each C_t^i contains only discriminants compatible to each other, whereas for any two discriminants d_1, d_2 such that $d_1 \in C_t^i, d_2 \in C_t^j, i \neq j$, we have $d_1 \# d_2$.

4.3.2 Context vectors spaces

A general idea for semantic translation, originated by Rapp (1999), consists in applying the rule that in similar target and source repositories the statistical distribution of collocations for the corresponding terms, source and its translation, should be similar to each other.

In order to explain how we build the vector space for the source repository D_S let us recall that a meaning $m \in \mathfrak{M}$ is represented by a set of compatible discriminants $C_t^i = \{d_1, \ldots, d_l\}$. Let us note that given $t \in D_S$, and C_t^i we get the set W_{tm} of all the termsets from R_S for the given meaning by performing searches in the repository index for the queries $t \wedge d_j$, for each $d_j \in C_t^i$. So, $W_{tm} = \{X : t \in X, d \in X, d \in C_t^i\}$. Having W_{tm} we filter out of W_{tm} only those termsets that contain some terms from the seed γ, building the set of W_{tm}^γ by translating x to $\gamma(x)$. So, we have the space $W^\gamma = \{W_{tm}^\gamma : t \in D_S \wedge m \in \mathfrak{M}\}$. For the target repository, we proceed in a similar way, although, as we do not induce senses, for each $t \in D_T$ a set of termsets V_t is built giving rise to the space $V = \{V_t : t \in D_T\}$, which then is filtered out to the space V^γ in such a way that only such termsets remain in V_t^γ, which contain some terms from $CoDom(\gamma)$.

Now for W^γ and V^γ two vector spaces can be built $\mathbf{S_S}$ and $\mathbf{S_T}$ for source and target termset spaces respectively. Clearly, the dimension of each space is limited to the seed size (i.e. $|Dom(\gamma)|$). Having the two spaces, for each $x \in \mathbf{S_S}$ we look for the n most similar $y \in \mathbf{S_T}$. For the similarity measure we use the $cosin$ pseudometrics (justified by the fact that the sets of termsets may be different in size), thus obtaining the translation candidates.

The question is though, how the vector spaces should be built, so that the similarity measure between x and y identifies properly y as the translation of x. In the paper by Koehn and Knight (2002) the vectors are built based on frequencies of "neighboring terms", i.e. in the vector corresponding to t, $i\text{-}th$ position represents frequency of x_i in all the windows with t. In our case the frequency vectors in $\mathbf{S_S}$ are built based on frequencies of context terms in W_{tm}^γ.

However, the frequency measure works "locally", namely, it does not take into account how often given termsets appear in the termset spaces for other pairs (t, m). Actually, one can expect that if a context x_i appears often within the frequent termsets of W_{tm}^γ but not too often within the other sets of frequent termsets, it plays a special role in identifying the pair (t, m), and a similar situation should repeat in the target language. And other way around, if x_i appears in too many sets $W_{t_p m_q}^\gamma$, its role in distinguishing given pair (t, m) decreases in both languages. It is actually an adoption of the well known measure $tf\text{-}idf$, which takes into account a distinctive value of particular context termsets. Therefore, in addition to the experiments with the frequency based vector spaces we have decided to build the vector spaces based on the $tf\text{-}idf$ measure. Below, the two methods of building vector spaces, one

based on the frequencies, and another one on $tf\text{-}idf$, are called briefly *local* and *global*, respectively. In the next section we compare the two approaches.

5 Experiments

The experiments have been performed for English (source) and Polish (target). We used two text corpora, built from the English and Polish Wikipedia versions respectively. The wikilinks between the articles, descriptive labels and interlingual links have been neglected. So, the Wikipedia versions were used only as sets of paragraphs. First, by applying the algorithm for finding translations based on the lexical similarities we have built a seed, then we have performed experiments with semantic translations. Below we present and discuss the experiment results.

Lexical similarity experiments The lexical similarity experiments are summarized on the Figs. 3 and 4. We have selected 5000 most frequent words from each corpus. Figure 3 shows the most frequent examples of the rules induced from D_S and D_T. As one can see (Fig. 4), from these starting dictionaries, 1507 translations have been properly detected, giving the precision 66,9 % (Fig. 4a). The precision can be increased by eliminating some "dummy rules", reducing slightly the recall. A good example of a "dummy rule" for English-to-Polish translation is $s \rightarrow \varnothing$, resulting from plural forms of English nouns translated to singular Polish equivalents. Just removing this one rule increases precision to 78,1 % (Fig. 4b). Even without this manual intervention the results outperform the results obtained by Koehn and Knight (2002) – with automatically generated rules we are able to find a larger seed, in spite of the fact that the source and target languages are from different language families, and the used corpora, although comparable, are not parallel as was in (Koehn and Knight 2002). It results from the fact that the procedure of building the rewriting rules discovers the ones which are statistically the most important for the corpora.

Fig. 3 *Top* frequent rules generated by lexical similarity translation

rule			support
c	→	k	213
l	→	∅	103
e	→	∅	69
v	→	w	59
y	→	ia	38
ph	→	f	35
∅	→	ny	30
all	→	zn	30
tion	→	cja	28
s	→	z	22

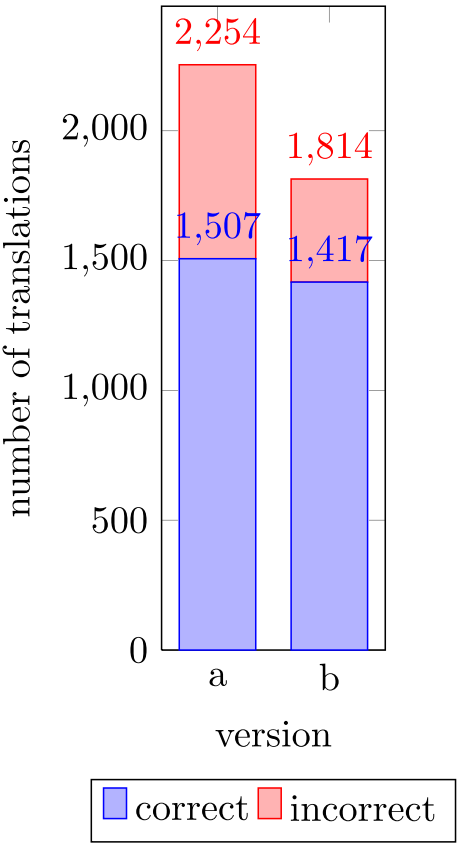

Fig. 4 Precision and recall of lexical similarity translation

Semantical similarity experiments Given the seed, we have performed series of experiments with semantic translations. The first experiments were focused on finding the ways of reducing the noise typical for vector representation. We have tested two possibilities:

1. reducing the influence of the noisy dimensions of the vector space
2. boosting the role of the most distinctive dimensions by applying a $tf\text{-}idf$ based measure for building the vector space.

For (1) we have tested a possibility of reducing the noise by zeroing in the vectors the less important dimensions. In particular, we have processed each vector in such a way that only the top n dimensions (i.e. the ones having the highest weights) are kept for the similarity calculations, whereas all the other dimensions are set to zero. Below, the procedure of zeroing the less valued dimensions is named *context vector noise reduction*. We say that n is $context\ size$ for the vector space. The procedure does not reduce the dimensionality of the vector spaces. Obviously, the lower is the value of the context size, the faster is the translation process. But what is more important, the experiments have shown that the noise reduction procedure not only reduces the computation time, but first of all up to a certain value of n it provides better results. One of our goals was to find out optimal values of the context size for particular cases of vector spaces.

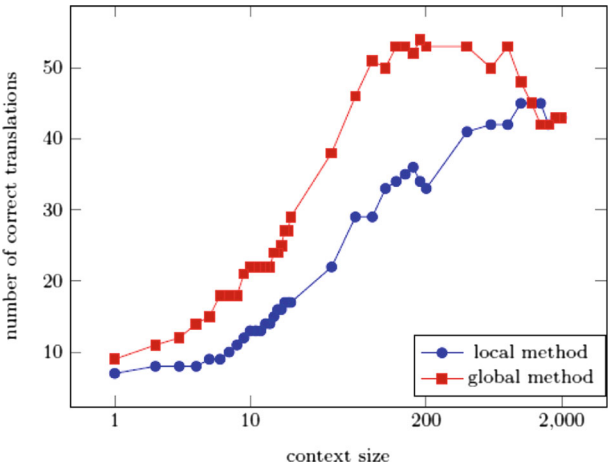

Fig. 5 Comparison of the local and the global versions with context vector noise reduction

For (2), we have compared the quality of the semantic similarity for the vector spaces with local dimension measures *versus* vector spaces with globally evaluated dimensions, i.e. frequency based measures versus $tf\text{-}idf$ ones.

From English Wikipedia we have selected manually 1000 terms having good translations in Polish Wikipedia[5] and single well defined meanings, i.e. the ones for which the algorithm SnS finds only one meaning. Figure 5 shows the results for the selected subset. The translations are considered as correct if the correct translation is among the first 10 translation candidates from D_T, i.e. the 10 candidates having the highest similarity measure to the source terms. From the experiment results the following observations can be made:

1. processing the vector spaces with the noise reduction procedure essentially reduces the computation cost of translation; up to a certain value n of the vector context size it also improves the translation quality of both, local and global methods;
2. vector spaces with globally evaluated dimensions provide better quality of the translations than the spaces with local dimensions; globally calculated vector spaces reach good translation quality already for the context size $n \in [100..200]$;
3. a similar quality of translation for local method can be reached for much higher values of the context size ($n \in [1000..2000]$)

In our previous paper (Krajewski et al. 2014), the vector space was built in a similar way as in (Koehn and Knight 2002), i.e. without the phase of identifying the set of meanings, so the context vectors were built just for t, instead of the pairs (t, m). The next series of experiments were devoted to the question how multiple meanings of the terms may deteriorate the quality of translations, and how discovering meanings and assigning them to the terms can improve the quality. To this end, in the second series of experiments we have selected manually a sample of 200 polysemic terms from D_S, for which the number of meanings in English Wikipedia was at least 5 and performed the following two tests:

[5]There are around 100000 terms in English Wikipedia having translations in Polish Wikipedia

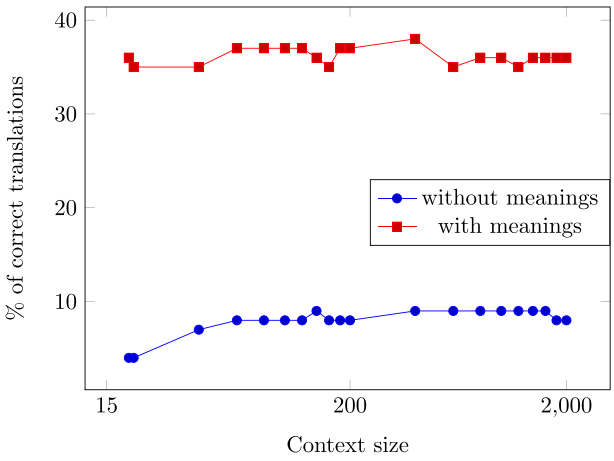

Fig. 6 Comparison of the translation quality for the polysemic terms

1. for the sample of highly polysemic terms we have build the vector space, and then tested the quality of translation without taking into account the meanings identified by the SnS phase;
2. for the same sample we have identified the meanings with SnS and then built the vector space for the pairs (t, m).

Figure 6 illustrates the results of the experiments. In both experiments the phase of the context vectors noise reduction have been performed, so that the quality of translations can be seen in the function of the context size parameter. As one can see the SnS phase of finding the meaning and then building the context vectors for the pairs (t, m) improves the quality of translations essentially. We can also see that for the vector space without meanings the

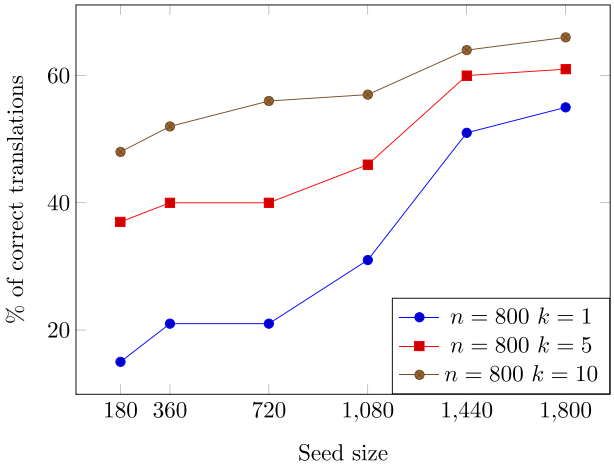

Fig. 7 Seed size influence for the translation quality of *top* $k = 1, 5, 10$ translation results for the vector context size $n = 800$

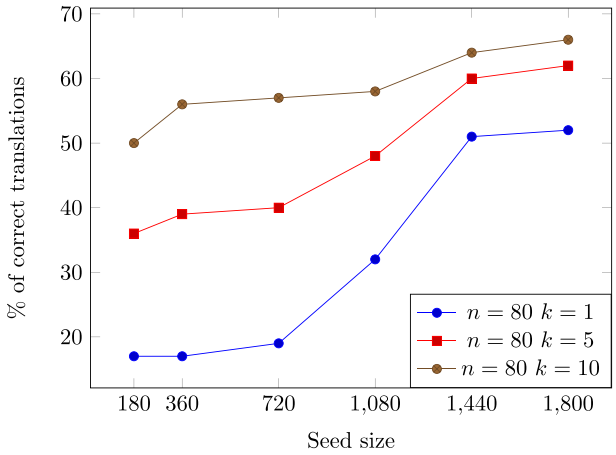

Fig. 8 Seed size influence for the translation quality of top $k = 1, 5, 10$ of translation results for the vector context size $n = 80$

optimal value for the context size is much higher than for the vector space with meanings (about 1000 versus 100).

Additionally, we have performed experiments aiming at checking how the size of the seed dictionary influences the semantic translation quality. The results are shown on the Figs. 7 and 8. The experiments included the SnS phases for the meaning detection. Figure 7 illustrates the experiments with the vector context size $n = 800$, whereas Fig. 8 shows the results for $n = 80$. As one can see, for the vectors built for pairs (t, m) comparable results can be obtained with smaller context size, which means that the algorithm quality is determined by the meaning injection. Still, there are limitations in obtaining very high precision. From the figures one can see though, that by increasing the number of k best candidates we can reach reasonable results.

6 Conclusion

In this work we proposed a novel approach to identify word translations from non-parallel or even unrelated texts. Comparing to the original seed based translation approach (Koehn and Knight 2002), the novel elements introduced in $SBDB^+$ are: (1) the phase of inducing lexicographical rules of translations; and (2) the phase of finding meanings of the terms from the dictionary.

The phase for finding meanings has been performed with the SnS method (see (Kozlowski and Rybinski 2014)).

According to our experiments, the size of the seed dictionary influences the quality of the phase of semantic translation. Therefore the proposed technique can be used iteratively, i.e. having discovered proper translations of the meanings in the semantic translation phase we can add the translations to the seed, and iteratively continue the translations for the expanded seed for those meaning that have not been positively translated earlier.

The proposed translation method is knowledge-poor and language independent. It is therefore applicable for maintaining multilingual ontologies devoted to continuously and dynamically changing domains. As shown, the method works well even for the languages

from different language families. We have also shown that the procedure of noise reduction of the context vectors space improves the translation precision and improves computational efficiency.

The referred evaluations (Rapp (1999), Koehn and Knight (2002)) were performed on small datasets (100-1000 words) extracted from a radio news corpus, or some existing lexicons. The results presented in this paper vary from 40 % to 72 % of correctness. Opposite to that, we performed evaluation on all words from wikipedia texts, used as a text repository. Our method reports 50 % of correct translations found as the first candidates, and up to 80 % among the top 10 candidates in the case of polysemic terms.

As future work we plan developing the syntactic part of our method by verifying meanings of the syntactic translations with the SnS algorithm. We also intend to incorporate the method to our knowledge base software Ω-Ψ^R, (Koperwas et al. 2014) for balancing indexes for multilingual full text information retrieval in English and Polish, i.e., giving similar results for queries regardless of the language.

References

Ballesteros, L., & Croft, W. (1997). Phrasal translation and query expansion techniques for cross-language information retrieval. In *ACM SIGIR forum*, Vol. 31 (pp. 84–91). ACM.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, *16*(2), 79–85.

Cimiano, P., Schultz, A., Sizov, S., Sorg, P., & Staab, S. (2009). Explicit versus latent concept models for cross-language information retrieval. In *IJCAI*, Vol. 9 (pp 1513–1518).

Deng, Y., & Byrne, W. (2008). Hmm word and phrase alignment for statistical machine translation. *IEEE Transactions on Audio, Speech, and Language Processing*, *16*(3), 494–507.

Dumais, S. T., Letsche, T. A., Littman, M. L., & Landauer, T. K. (1997). Automatic cross-language retrieval using latent semantic indexing. In *AAAI spring symposium on cross-language text and speech retrieval*, Vol. 15 (p 21).

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, Vol. 7 (pp. 1606–1611).

Gracia, J., Montiel-Ponsoda, E., Cimiano, P., Gómez-Pérez, A., Buitelaar, P., & McCrae, J. (2012). Challenges for the multilingual web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, *11*, 63–71.

Grefenstette, G. (1993). Evaluation techniques for automatic semantic extraction: comparing syntactic and window based approaches. In *Proceedings of the SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*, Columbus Ohio.

Harris, Z.S. (1981). *Distributional structure*. Springer.

Hotho, A., Maedche, A., Staab, S., Zacharias, V., et al. (2003). On knowledgeable unsupervised text mining. *Text Mining*, 131–152.

Hull, D.A., & Grefenstette, G. (1996). Querying across languages: a dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 49–57). ACM.

Koehn, P., & Knight, K. (2002). Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition-Volume 9, Association for Computational Linguistics* (pp. 9–16).

Koehn, P., Och, F.J., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American chapter of the association for computational linguistics on human language technology* Vol. 1 (pp. 48–54). Association for Computational Linguistics.

Koperwas, J., Skonieczny, L., Kozlowski, M., Andruszkiewicz, P., Rybinski, H., & Struk, W. (2014). AI platform for building university research knowledge base, In Andreasen, T., Christiansen, H., Cubero, J. C., & Ras, Z. W. (Eds.), *Foundations of Intelligent Systems*, Vol. 8502: LNCS, Springer, Springer International Publishing, Lecture Notes in Artificial Intelligence.

Kozlowski, M. (2014). Word Sense Discovery using Frequent Termsets. PhD Thesis, Warsaw University of Technology.

Kozlowski, M., & Rybinski, H. (2014). Sns: A novel word sense induction method, In Kryszkiewicz, M., Cornelis, C., Ciucci, D., Medina-Moreno, J., Motoda, H., & Ras, Z.W. (Eds.), *Rough Sets and Intelligent Systems Paradigms*, (Vol. 8537 pp. 258–268): Proceedings, Springer, LNAI.

Krajewski, R., Rybinski, H., & Kozlowski, M. (2014). A seed based method for dictionary translation. In *Foundations of intelligent systems*, Vol. 8502 (pp. 415–424). LNCS, Springer.

Kranich, S., Becher, V., & Höder, S. (2011). A tentative typology of translation-induced language change. *Kranich, Becher, Höder and House*, 11–44.

McCrae, J., Montiel-Ponsoda, E., Aguado de Cea, G., Espinoza Mejía, M.J., & Cimiano, P. (2011). Combining statistical and semantic approaches to the translation of ontologies and taxonomies.

McMahon, A.M. (1994). *Understanding language change*: Cambridge University Press.

Mihalcea, R., & Moldovan, D.I. (2001). Ez. wordnet: Principles for automatic generation of a coarse grained wordnet. In *FLAIRS conference*, (pp. 454–458).

Miller, G.A., Chodorow, M., Landes, S., Leacock, C., & Thomas, R.G. (1994). Using a semantic concordance for sense identification. In *Proceedings of the workshop on human language technology, Association for Computational Linguistics* (pp. 240–243).

Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250.

Nykiel, T., & Rybinski, H. (2008). Word sense discovery for web information retrieval. In *Data mining workshops*, 2008. IEEE International Conference on ICDMW'08 (pp. 267–274). IEEE.

Pimienta, D., Prado, D., & Blanco, Á. (2009). Twelve years of measuring linguistic diversity in the Internet: balance and perspectives. United Nations Educational and Scientific and Cultural Organization.

Pirkola A (1998). The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 55–63). ACM.

Protaziuk, G., Wróblewska, A., Bembenik, R., & Podsiadły-Marczykowska, T. (2012). Lexo: a lexical layer for ontologies–design and building scenarios. *Studia Informatica*, *33*(2B), 173–186.

Rapp, R. (1995). Identifying word translations in non-parallel texts. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics* (pp. 320–322).

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, association for computational linguistics* (pp. 519–526).

Rybinski, H., Kryszkiewicz, M., Protaziuk, G., Kontkiewicz, A., Marcinkowska, K., & Delteil, A. (2008). Discovering word meanings based on frequent termsets. *Mining Complex Data*, 82–92.

Salton, G. (1970). Automatic processing of foreign language documents. *Journal of the American Society for Information Science*, *21*(3), 187–194.

Soergel, D. (1997). Multilingual thesauri in cross-language text and speech retrieval. In *AAAI Symposium on cross-language text and speech retrieval, Citeseer* (pp. 24–26).

Sorg, P., & Cimiano, P. (2008a). Cross-lingual information retrieval with explicit semantic analysis.

Sorg, P., & Cimiano, P. (2008b). Enriching the crosslingual link structure of wikipedia-a classification-based approach. In *Proceedings of the AAAI 2008 workshop on wikipedia and artifical intelligence* (pp. 49–54).

Vogel, S., Ney, H., & Tillmann, C. (1996). Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on computational linguistics-volume 2, Association for computational linguistics* (pp. 836–841).

Wróblewska, A., Podsiadły-Marczykowska, T., Bembenik, R., Protaziuk, G., & Rybiński, H. (2012). Methods and tools for ontology building, learning and integration–application in the synat project. In *Intelligent tools for building a scientific information platform* (pp. 121–151). Springer.

Yamada, K., & Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of the 39th annual meeting on association for computational linguistics, association for computational linguistics* (pp. 523–530).