Bilingual dictionaries for all EU languages

Ahmet Aker, Monica Lestari Paramita, Mārcis Pinnis*, **Robert Gaizauskas**
Department of Computer Science, University of Sheffield, UK
*Tilde, Vienibas gatve 75a, Riga, Latvia, LV-1004
Email: a.aker@dcs.shef.ac.uk, m.paramita@sheffield.ac.uk
marcis.pinnis@tilde.lv, r.gaizauskas@dcs.shef.ac.uk

Abstract

Bilingual dictionaries can be automatically generated using the GIZA++ tool. However, these dictionaries contain a lot of noise, because of which the qualities of outputs of tools relying on the dictionaries are negatively affected. In this work, we present three different methods for cleaning noise from automatically generated bilingual dictionaries: LLR, pivot and transliteration based approach. We have applied these approaches on the GIZA++ dictionaries – dictionaries covering official EU languages – in order to remove noise. Our evaluation showed that all methods help to reduce noise. However, the best performance is achieved using the transliteration based approach. We provide all bilingual dictionaries (the original GIZA++ dictionaries and the cleaned ones) free for download. We also provide the cleaning tools and scripts for free download.

Keywords:: GIZA++ dictionaries, EU languages, dictionary cleaning

1. Introduction

Bilingual dictionaries are important for various applications of human language technologies, including cross-language information search and retrieval, machine translation and computer-aided assistance to human translators. The GIZA++ (Och and Ney, 2000; Och and Ney, 2003) tool provides an automated way to construct bilingual dictionaries from parallel corpora. However, there are two main problems using this tool to create such bilingual dictionaries.

The first problem is that the tool is hard to use and input data preparation is difficult. For technically non-sophisticated users, installing and running GIZA++ is not at all straightforward. Depending on the level of technical ability of the installer, setting the tool up can take several weeks to finish successfully. Additionally, preparation of parallel data to be input to the tool is also time consuming, as any input to GIZA++ must be pre-processed to satisfy certain conditions. Data preparation time is increased if the aim is to generate bilingual dictionaries for many languages.

The second problem has to do with noise in the automated bilingual dictionaries. GIZA++ treats every word in the source language as a possible translation for every word in the target language and assigns the pairs probabilities indicating the likelihood of the translations. A word pair with lower probability can be regarded as an “incorrect” translation and a word pair with higher probability as a “correct” translation. However, this is an ideal situation and is not always the case in GIZA++ as pairs of words with high translation probabilities may still be wrong. Due to this problem, any application that makes use of word pair translations only above a probability value threshold is still served with noise. Aker et al. (2012), for instance, use GIZA++ dictionaries as a feature when extracting parallel phrases from comparable corpora and report mistranslated pairs of phrases mainly due to noise in the statistical dictionaries. Although the authors clean their dictionaries by removing every entries that have lower probability values than a manually determined threshold, their results show that better cleaning is required. The best way to do this would be to manually filter out all wrong translations. However, this is a labour intensive task, which is not feasible to perform for many language pairs. Another alternative would be an automated approach that, unlike Aker et al. (2012), does not trivially delete all dictionary entries below a probability threshold but instead aims to filter out mistranslations independently from any manually set threshold.

In this paper we address both problems. To address the first problem, we pre-generate bilingual dictionaries for all official European languages except Irish and Croatian and provide them for free download. To address the second problem, we describe three different cleaning techniques, two of which are novel. We apply these cleaning techniques on the statistical dictionaries in order to reduce noise. Thus the data we offer for downloading contains several versions of the same bilingual dictionaries – the original GIZA++ output and multiple cleaned versions. We also provide access to our cleaning methods in the form of open source tools for natural language processing-based system developers.

In the remainder of the paper, we first describe the data we use to generate the bilingual dictionaries (Section 2). Next, we introduce our cleaning methods (Section 3). In Section 4, we describe our evaluation set-up and provide results that were acquired by performing a manual quality evaluation of the cleaning processes. Section 5 lists the resources that are available for download. Finally, we conclude the paper with Section 6.

2. Bilingual dictionaries

To obtain the original GIZA++ dictionaries we used the freely available DGT-TM parallel corpus (Steinberger et al., 2012), which provides data for official languages of the European Union. The number of sentence pairs available for each language pair is shown in Table 1.

As shown in Table 1, the number of sentence pairs available in the DGT-TM corpora varies between language pairs, ranging from under 1.8M for RO-EN to over 3.7M for the

Language Pair	Sentence Pairs
EN-BG	1,810,612
EN-CS	3,633,782
EN-DA	3,179,359
EN-DE	3,207,458
EN-EL	3,016,402
EN-ES	3,175,608
EN-ET	3,652,963
EN-FI	3,135,651
EN-FR	3,692,787
EN-HU	3,789,650
EN-IT	3,221,060
EN-LT	3,736,907
EN-LV	3,722,517
EN-MT	2,130,282
EN-NL	3,164,924
EN-PL	3,665,112
EN-PT	3,620,006
EN-RO	1,781,306
EN-SK	3,721,620
EN-SL	3,689,972
EN-SV	3,248,207

Table 1: DGT-TM parallel data statistics

following language pairs: EN-HU, EN-LT, EN-LV and EN-SK. On average, each language pair contains 3.2M sentence pairs. Using these data, we created bilingual dictionaries for 21 language pairs. We exclude English-Irish because the amount of parallel data available in DGT-TM is very small. We exclude also English-Croatian, because DGT-TM does not cover Croatian.

Each bilingual dictionary entry has the form $\langle s, t_i, p_i \rangle$, where s is a source word, t_i is the i-th translation of s in the dictionary and p_i is the probability that s is translated to t_i, the p_i's summing to 1 for each s in the dictionary.

We use these original dictionaries and run our cleaning methods on them to remove noise. These methods are the subject of the next section.

3. Methods

To clean the GIZA++ dictionaries described in Section 2, we apply three different methods as described below.

3.1. Statistical approach

The first method we implement is similar to the one reported in Munteanu and Marcu (2006). The method uses LogLikelihood-Ratio (LLR) (Dunning, 1993) as a test to decide whether a pair of source and target words are correct or incorrect translations of each other. Any pair not passing the test is filtered from the dictionary.

To do this, we first align the parallel sentence pairs using the GIZA++ toolkit (Och and Ney, 2000; Och and Ney, 2003) in both directions and then refine the alignments using a “grow-diag-final-and” strategy. The grow-diag-final-and entails for each sentence pair the alignment information between the words. Based on this alignment file we construct the co-occurrence matrix used to compute the LLR:

$$Co - occurenceMatrix = \begin{array}{ccc} & T & T' \\ S & k11 & k12 \\ S' & k21 & k22 \end{array}$$

where S is the source word and T is the target word. The aim is to assess whether S and T are translations of each other. S' and T' represent source and target words other than S and T. The entry $k11$ is the number of times S and T occurred together (aligned to each other), $k12$ is the number of times T occurred with S', $k21$ is the number of times S occurred with T', and $k22$ is the number of times S' and T' occurred together. We filter out any pair in the GIZA++ dictionary whose LLR value was below 10.83 ($p < 0.001$) by looking at the χ^2 significance table.

Note that we also skip dictionary entries which start or end with punctuations or symbols. Furthermore, we also delete any dictionary entry whose GIZA++ probability is below 0.001. These filters are applied regardless of the χ^2 statistics.

3.2. Transliteration based approach

The second method we have investigated tries to use a transliteration-based approach on filtering the dictionaries. The idea is that simply applying thresholds on probabilistic dictionaries will filter out also good translation equivalents. However, identification of translation equivalents that can be transliterated from one language to the other may allow identifying good pairs below the applied thresholds and thus keep them in the filtered dictionaries. The method filters dictionary entries using the following 7 steps:

1. The first step performs dictionary entry structural validation in order to remove obvious noise. At first, we remove all entries that contain invalid character sequences on either source or target side. Character sequences are considered invalid if according to the Unicode character table they contain control symbols, surrogate symbols or only whitespace symbols. In this step we also identify mismatching character sequences by comparing the source and target sides of a dictionary entry. At first we verify that the source and target token letters are equally capitalised (with an exception of the first letter, which in some languages, e.g., for nouns in German or days of a week in English, is capitalised). Further, we verify whether the letters contained in the source and target sides belong to the source and target language alphabets and whether both tokens contain equal numbers of digits, punctuation marks, and symbols, and whether they are located in similar positions in the source and target words. As the GIZA++ probabilistic dictionaries are statistical representations of token alignments in a parallel corpus, the alignments contain also easily detectable mistakes, such as, words paired with punctuations, incorrectly tokenized strings paired with words, etc. By applying character-based validation rules on the source and target language words we can easily filter out such obvious mistakes in the probabilistic dictionaries.

2. The second step identifies dictionary entries that are transliterations. We apply two different transliteration methods: 1) the language independent (how-

Source Token	Target Token	GIZA++ Probability	Filtering Step
.	94/65/ek.	0.50	Structural validation (1) - wrong entries
standards	standarts	0.02	Transliteration identification (2) - correct entries
a	aprobēt	0.50	IDF score-based filter (3) - wrong entries
proven	gazprom	0.08	Threshold filter (4) - wrong entries
regulatory	energoregulatora	0.50	Partial containment and transliteration filter (5) - wrong entries
navigational	dodamos	1.00	Heuristic filters (6) - wrong entries

Table 2: English-Latvian dictionary entries identified according to different filtering steps

ever, fixed to the Latin, Greek, and Cyrillic alphabets) rule-based transliteration method proposed by Pinnis (2013), which transliterates words into English using simple letter substitution rules, and 2) the character-based statistical machine translation method also proposed by Pinnis (2013). While the first transliteration method is fast, it is not able to capture morphological variations in different languages and it treats each character independently of the context. The second method, however, takes context (character n-grams) into account and is able to transliterate words not only into English, but also to other languages, thus transliterated word identification can be performed bi-directionally (from source to target and from target to source). In order to identify transliterated words, the transliterations (e.g., the source word transliterated into the target language) are compared with the other sides word (e.g., the target language word) using a string similarity measure based on the Levenshtein distance (Levenshtein, 1966). If the maximum similarity score using any of the transliteration methods and directions (source-to-target or target-to-source) is higher than 0.7 (identified as an acceptable threshold through empirical analysis) and the source and target words are not equal (because such pairs are often wrong language pairs), we consider the dictionary entry as transliterated and we pass it through to the filtered dictionary (the further filtering steps are skipped).

3. In the third step we analyse the remaining pairs using reference corpora based inverse document frequency (IDF) scores (Jones, 1972) of the source and target words. We remove all pairs that have a difference of word IDF scores greater than 0.9 (also empirically identified). Such pairs often indicate of functional word (or stop-word) miss-alignment with content words (e.g., in the dictionaries the English “a” is usually paired with almost everything else and the IDF-based filter reliably removes such entries).

4. In the fourth step we apply a translation probability value threshold that is differentiated for (source language) words that were already containing transliteration pairs (i.e., if a dictionary entry containing the source word was identified as a transliteration, then all other translation candidates for the source word are required to have a high probability in order to be accepted as translation equivalents).

5. Then, we remove all pairs that partially contain transliterations. For instance, consider the dictionary entry “monopoly” (in English) and “monopols” (in Latvian). The entry is a transliteration, thus, “monopolsituācijā” (translated as “in the case of a monopoly” would be filtered out as it contains fully the transliterated part.

6. We apply also several heuristic filters that have shown to remove further noise (e.g., rare words miss-aligned with a probability of one if a source word already contains multiple translation hypotheses, equal source and target words if the source word already contains multiple translation hypotheses, etc.).

7. Finally, the pairs that have passed all filter tests are written to the filtered dictionary.

Examples of dictionary entries that were identified using the different filtering steps from the English-Latvian GIZA++ dictionary are given in Table 2.

3.3. Pivot language based approach

The pivot language based approach uses the idea of intermediate languages to clean noise from the bilingual dictionaries. The idea of a pivot language is used in related work to overcome the problem of unavailable bilingual dictionaries such as in cross lingual information retrieval (CLIR) (Gollins and Sanderson, 2001; Ballesteros, 2002), in statistical machine translation (Wu and Wang, 2007; Wu and Wang, 2009) and bilingual dictionary generation (Paik et al., 2001; Seo and Kim, 2013). However, our approach differs from related work by adopting the idea of pivot languages to clean noise from existing dictionaries instead of using it for translation purposes. This means that we aim at cleaning an existing dictionary such as for the English-German language pair using intermediate dictionaries such as German-French and French-English. In this case, the pivot language is French.

Our approach uses the bilingual dictionary that has to be cleaned as the starting point. In Figure 1, this is the English-German (EN-DE) GIZA++ dictionary. We distinguish between one pivot language and several parallel pivot languages approach. In the one pivot language approach (shown as the blue arrow in Figure 1), our method takes for every source language (i.e., English) word enW its translations in the target (i.e., German) language ($deW_1, ..., deW_n$). In the next step, using a DE-FR GIZA++ dictionary, each such German word, deW_i, is then translated into French leading to French words

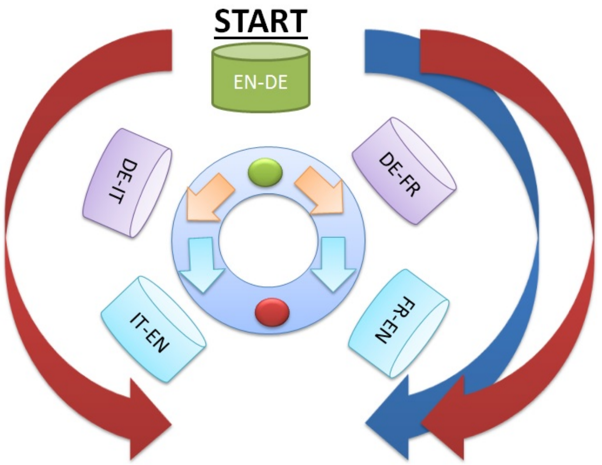


Figure 1: Pivot language based approach.

$frW_{i1}, ..., frW_{im}$. Each such French word, frW_{ij}, is then looked up in an FR-EN GIZA++ dictionary leading to possible translations in English ($enW_{ij1}, ..., enW_{ijp}$). If none of the English words $enW_{ij1}, ..., enW_{ijp}$ matches enW then the pair $< enW, deW_i >$ is removed from the EN-DE dictionary.

Our early experiments showed that using the one pivot language approach many entries from the EN-DE dictionary are removed because the pivot dictionary (DE-FR) does not contain entries for the German words. To overcome this problem we also introduce the several parallel pivot languages approach (shown as the red arrows in Figure 1) where instead of using one pivot language, we perform the cleaning with two pivot dictionaries at the same time. That means when we perform the cleaning of EN-DE using DE-FR-EN (as described in the one pivot language approach), we also perform in parallel the cleaning using another pivot dictionary, such as DE-IT-EN. In Figure 1 the two parallel pivot languages approach is shown using DE-FR-EN and DE-IT-EN. If at least one of these returns an English word enW_{ijp} equal to enW we keep the entry $< enW, deW_i >$ in the EN-DE dictionary otherwise the entry is removed. By performing two parallel checks we reduce the chance that the entry $< enW, deW_i >$ is removed from the dictionary because of missing entries.

Note that similarly to the LLR method within this approach we also skip –independently from the pivot language dictionary look-ups– dictionary entries which contain punctuations or symbols and also entries whose dictionary probability values are below 0.001.

4. Evaluation

To assess the performance of the different cleaning methods we performed a manual evaluation task by asking humans to judge the translation quality of the remaining dictionary entries. In the evaluation we randomly selected dictionary entries from 8 different sets to assess. The sets are shown in Figure 2. The first set contains all entries from the original GIZA++ dictionary, which do not appear in any of the other 7 sets (i.e. they are not retrieved by any of the three approaches). This set is used to understand whether the cleaning methods do miss good data or not. The next four sets are the entries in the intersections between the results of the three methods, *I-1*, *I-2*, *I-3* and *All*. The *All* set contains only entries which are also found in the other results. The other intersection sets contain entries between two methods. Finally, we have the *LLR*, *Pivot* and *Transliteration* sets, which do not share any entry with the intersection sets. Figure 3 shows the number of dictionary entries in each of the 8 sets for the English-German language pair. For instance, the *Pivot* method outputs for the English-German dictionary in total 277,703 entries. However, we divide this set into 4 parts: portion within *All* intersection (8,987 entries), portion of entries which intersects with the *LLR* method (*I-1*, in total 91,924 entries), portion intersecting with the *Transliteration* based method (*I-3*, in total 20,605 entries) and finally what is distinct within the *Pivot* result set (in total 156,187 entries).

From each of the 8 sets, we randomly selected 40 entries leading to total 320 entries and showed them to human assessors. Each assessor judged all 320 entries. In the assessment, similar to Aker et al. (2013), we asked human asses-

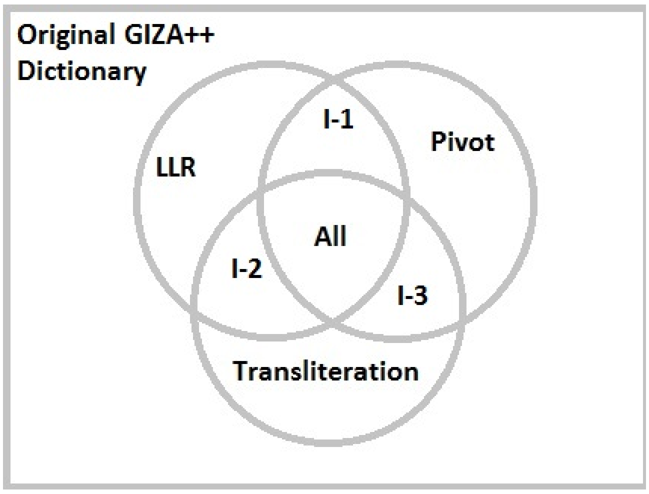


Figure 2: Evaluation sets.

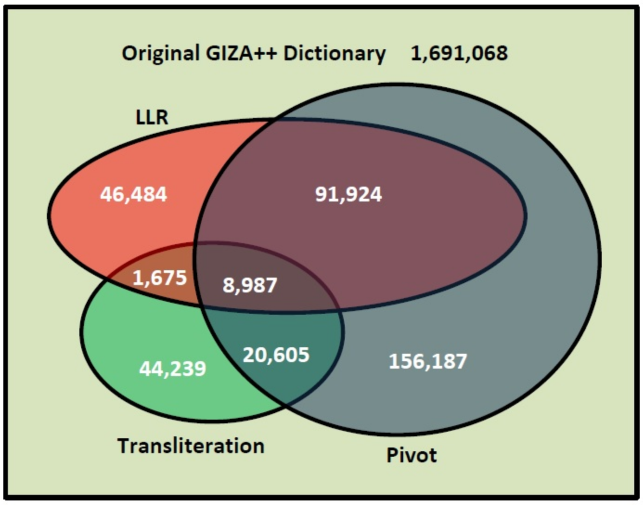


Figure 3: Number of entries in each set for English-German.

sors to categorize each presented dictionary entry into one of the categories shown in Figure 4. Two German and two Latvian native speakers who were fluent in English took part in this evaluation task. Note that in the evaluation we only used the English to X (i.e. German and Latvian) dictionaries. However, we also provide cleaned version of the dictionaries from language X to English.

4.1. Results

The results of the evaluation are shown in Table 3 for English-German and Table 4 for English-Latvian.

From the results we can see that the dictionary entries from the original GIZA++ dictionary are very noisy. Only 2%-6% of the entries contain correct translations. Note that these entries are not included in any of the cleaned sets. This means that the cleaning methods are good filters to skip such noisy entries. Furthermore, the results show that the transliteration method performs best compared to the other two cleaning methods for both English-German and English-Latvian language pairs. According to the manual assessors this method achieves around 55%-61% precision. The pivot approach achieves around 40%-42% for both language pairs. The LLR method gets for English-German only 20%, but for the English-Latvian language pair it achieves a similar figure as the pivot approach. However, these figures are based on the entries not included in the intersection sets. If we look at the intersection sets we see that the precision figures go higher. If *All* intersections are considered then the precision results are just below 90% for both English-German and English-Latvian language pairs. Among the intersection sets the lowest precision results are achieved when the pivot method is intersected with the LLR approach (set *I-1*). The high precision scores in the intersection sets show that the cleaning methods commonly identify "good" translations and the highest figure in the *All* set suggests to combine the different cleaning methods and apply them together on the original GIZA++ dictionaries.

We also computed the agreement rates between the assessors. The German assessors had an agreement in 79.69% of all evaluated dictionary entries and the Latvian assessors agreed in 80.31% of all entries. We computed the agreement based on the number of agreed votes over the three categories and the 8 sets (see the second half of the Tables 3 and 4) divided by the total number of votes (in this case 320).

5. Resources for download

We have prepared the dictionaries as well as the cleaning methods for download:

- **Original GIZA++ dictionaries**: These are the dictionaries we obtained using the GIZA++ alignment tool. We do not apply any cleaning technique on these statistical dictionaries. The dictionaries can be found here: http://www.taas-project.eu/. For the purpose of the pivot approach we also created GIZA++ dictionaries for *DE-XX* and *FR-XX* where *XX* represents any of the other languages. These dictionaries can also be downloaded from the same link.
- **Cleaned bilingual dictionaries**: These are the cleaned versions of the original dictionaries. These dictionaries are also available through the same link as the original ones.
- **Tools and scripts for cleaning**: The LLR and the pivot approaches can be downloaded from: http://staffwww.dcs.shef.ac.uk/people/A.Aker/activityNLPProjects2.html. The transliteration-based cleaning tool's source code can be downloaded from: https://github.com/pmarcis/dict-filtering.

6. Conclusion

In this paper we have described three different methods for cleaning bilingual dictionaries: LLR, pivot, and the transliteration based approach. We have applied these methods on GIZA++ dictionaries covering 22 official EU languages. We also performed manual evaluation using English-German and English-Latvian dictionaries. Our evaluation shows that all methods help reducing noise, i.e., the dictionary entries not taken by the three methods are mainly judged by the assessors as noise. The best performance is achieved using the transliteration approach. We have also seen that the results in the intersection sets were higher than in the other sets. This showed that the cleaning methods do commonly identify what is a correct translation. We provide all bilingual dictionaries (the original GIZA++ dictionaries and the cleaned versions) free for download. We also provide the cleaning tools and scripts for free download.

For future work we aim to combine the different approaches using some machine learning techniques and apply them together on the cleaning task. Furthermore, we plan to work on other language pairs where English is not involved and provide them for free download. We plan to upload any additional dictionary to http://www.taas-project.eu/.

7. Acknowledgments

The research within the project TaaS leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013), Grant Agreement no 296312. We would like to thank the assessors who took part in our manual evaluation.

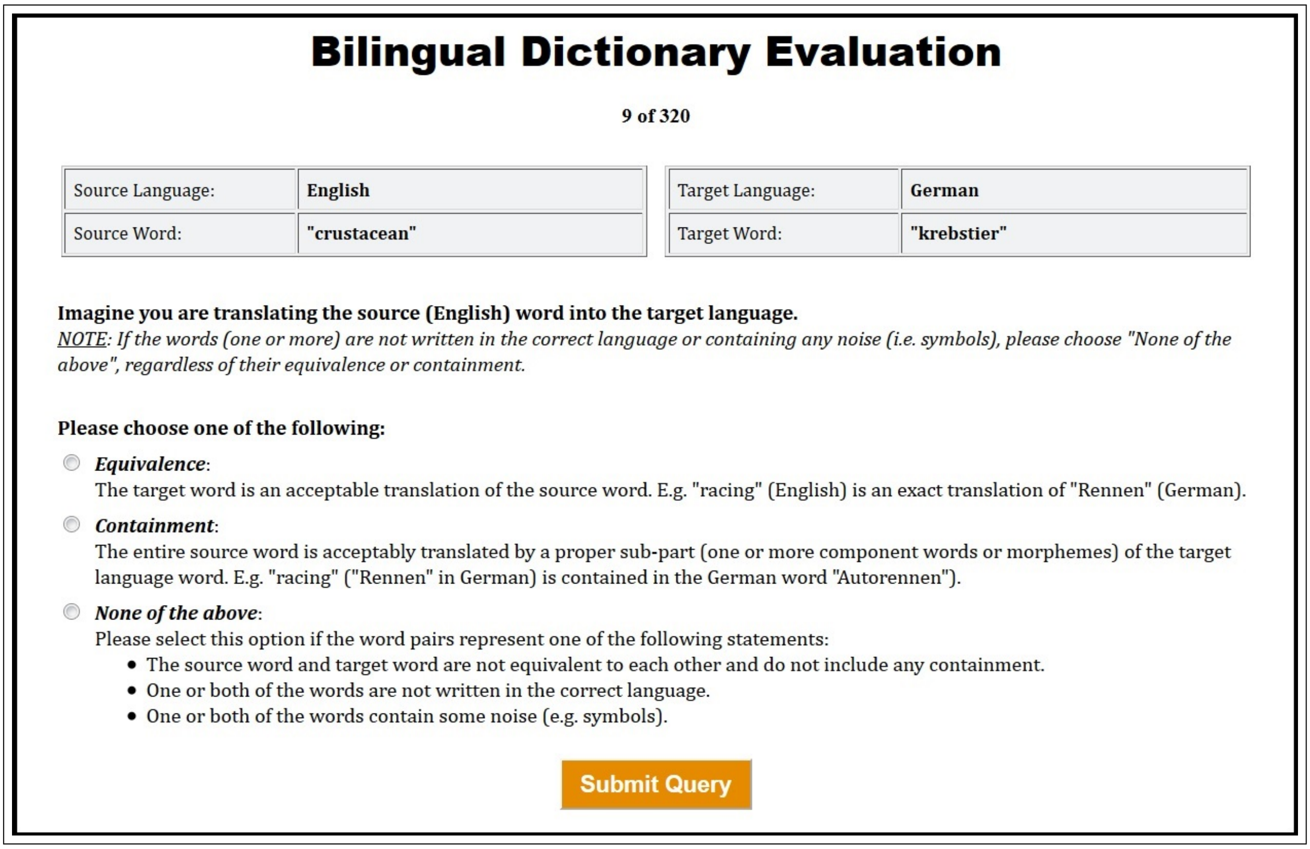
8. References

Aker, A., Feng, Y., and Gaizauskas, R. (2012). Automatic bilingual phrase extraction from comparable corpora. In *24th International Conference on Computational Linguistics (COLING 2012), IIT Bombay, Mumbai, India, 2012*. Association for Computational Linguistics.

Aker, A., Paramita, M., and Gaizauskas, R. (2013). Extracting bilingual terminologies from comparable corpora. In *The 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria)*. Association for Computational Linguistics.

Ballesteros, L. A. (2002). Cross-language retrieval via transitive translation. In *Advances in information retrieval*, pages 203–234. Springer.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74.



Bilingual Dictionary Evaluation

9 of 320

Source Language:	**English**	Target Language:	**German**
Source Word:	**"crustacean"**	Target Word:	**"krebstier"**

Imagine you are translating the source (English) word into the target language.
NOTE: If the words (one or more) are not written in the correct language or containing any noise (i.e. symbols), please choose "None of the above", regardless of their equivalence or containment.

Please choose one of the following:

○ ***Equivalence***:
The target word is an acceptable translation of the source word. E.g. "racing" (English) is an exact translation of "Rennen" (German).

○ ***Containment***:
The entire source word is acceptably translated by a proper sub-part (one or more component words or morphemes) of the target language word. E.g. "racing" ("Rennen" in German) is contained in the German word "Autorennen").

○ ***None of the above***:
Please select this option if the word pairs represent one of the following statements:
- The source word and target word are not equivalent to each other and do not include any containment.
- One or both of the words are not written in the correct language.
- One or both of the words contain some noise (e.g. symbols).

Figure 4: Bilingual dictionary evaluation set-up.

SetName	Eq.	Cont.	Wrong	Precision	Eq.	Cont.	Wrong	Precision
All	71	6	3	88.75%	35	2	0	94.59%
I-1	38	26	16	47.50%	15	12	4	48.39%
I-2	65	6	9	81.25%	30	2	2	88.24%
I-3	53	10	17	66.25%	21	2	4	77.78%
Transliteration	45	7	28	56.25%	19	2	11	59.38%
Pivot	32	22	26	40.00%	12	7	10	41.38%
LLR	16	28	36	20.00%	5	11	15	16.13%
Original	2	26	52	2.50%	0	10	24	0.00%

Table 3: Results of the EN-DE manual evaluation by two annotators. The second half of the table shows figures where we ignored the cases for which there was a disagreement. The precision figure in each row is computed by dividing the figure in column *Eq* with the sum of the figures of the columns *Eq* to *Wrong* of that row.

Gollins, T. and Sanderson, M. (2001). Improving cross language retrieval with triangulated translation. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 90–95. ACM.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707.

Munteanu, D. S. and Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 81–88, Morristown, NJ, USA. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics*, pages 1086–1090, Morristown, NJ, USA. Association for Computational Linguistics.

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

Paik, K., Bond, F., and Satoshi, S. (2001). Using multiple pivots to align korean and japanese lexical resources. In *Proc. of the NLPRS-2001 Workshop on Language Resources in Asia*, pages 63–70.

Pinnis, M. (2013). Context Independent Term Mapper for European Languages. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2013)*, pages 562—-570, Hissar, Bulgaria.

SetName	Eq.	Cont.	Wrong	Precision	Eq.	Cont.	Wrong	Precision
All	71	2	7	88.75%	33	0	1	97.06%
I-1	55	5	20	68.75%	25	2	7	73.53%
I-2	62	4	14	77.50%	25	0	3	89.29%
I-3	56	7	17	70.00%	25	1	6	78.13%
Transliteration	49	4	27	61.25%	21	0	11	65.63%
Pivot	34	11	35	42.50%	14	2	14	46.67%
LLR	34	4	42	42.50%	15	1	19	42.86%
Original	5	3	72	6.25%	0	0	32	0%

Table 4: Results of the EN-LV manual evaluation by two annotators. The second half of the table shows figures where we ignored the cases for which there was a disagreement. The precision figure in each row is computed by dividing the figure in column *Eq* with the sum of the figures of the columns *Eq* to *Wrong* of that row.

Seo, H.-S. K. H.-W. and Kim, J.-H. (2013). Bilingual lexicon extraction via pivot language and word alignment tool. *ACL 2013*, page 11.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., and Schlter, P. (2012). Dgt-tm: A freely available translation memory in 22 languages. In *Proceedings of LREC*, pages 454–459.

Wu, H. and Wang, H. (2007). Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181.

Wu, H. and Wang, H. (2009). Revisiting pivot language approach for machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 154–162. Association for Computational Linguistics.