Dutch Compound Splitting for Bilingual Terminology Extraction

Lieve Macken and Arda Tezcan

Ghent University, Department of Translation, Interpreting and Communication (Lieve.Macken@Ugent.be, Arda.Tezcan@Ugent.be)

Abstract

Compounds pose a problem for applications that rely on precise word alignments such as bilingual terminology extraction. We therefore developed a state-of-the-art hybrid compound splitter for Dutch that makes use of corpus frequency information and linguistic knowledge. Domain-adaptation techniques are used to combine large out-of-domain and dynamically compiled in-domain frequency lists. We perform an extensive intrinsic evaluation on a Gold Standard set of 50,000 Dutch compounds and a set of 5,000 Dutch compounds belonging to the automotive domain.

We also propose a novel methodology for word alignment that makes use of the compound splitter. As compounds are not always translated compositionally, we train the word alignment models twice: a first time on the original data set and a second time on the data set in which the compounds are split into their component parts. The obtained word alignment points are then combined.

The resulting word alignments are integrated in the TExSIS bilingual terminology extraction system and we show that the compound splitter combined with the novel word alignment technique considerably improves the bilingual terminology extraction results.

Keywords

Compound splitting, bilingual terminology extraction, word alignment, multi-word units, translation, Dutch

Introduction

Compounding is a highly productive process in Dutch that poses a challenge for various NLP applications that rely on automated word alignment such as machine translation and bilingual terminology extraction.

In Dutch, a compound is usually not separated by means of white space characters and hence constitutes one single word. Examples are *slaap+zak* (En: *sleeping bag*), *hoofd+pijn* (En: *head+ache*) and *[post+zegel]+verzamelaar* (En: *stamp collector*). Compounds written as one word are problematic for statistical word alignment as on the one hand they drastically increase the vocabulary size and on the other hand lead to one-to-many word alignments, which are more difficult to model as is the case in *slaapzak*, which corresponds to two words and *regeringshoofd*, which corresponds to three words in English (En: *head of government*).

Numerous studies showed that splitting compounds prior to translation model training improves the translation quality of statistical machine translation systems (Fritzinger & Fraser, 2010; Koehn & Knight, 2003; Stymne & Holmqvist, 2008). However, the impact of compound splitting on bilingual terminology extraction is less studied.

Most compound splitting approaches are corpus-based and use corpus frequencies to find the optimal split points of a compound (Koehn & Knight, 2003). Adding linguistic knowledge in the form of part-of-speech restrictions (Stymne & Holmqvist, 2008) or morphological information (Fritzinger & Fraser, 2010) reduces the number of erroneous split points.

As terminology extraction systems typically work with much smaller corpora than the training corpora of Machine Translation Systems, and as the accuracy of the compound splitter depends on the size and the quality of training corpus, we trained a stand-alone data-driven compound splitting tool on the basis of a frequency list derived from Wikipedia. The tool determines a list of eligible compound constituents (so-called heads and tails) on the basis of word frequency information and uses part-of-speech (PoS) information as a means to restrict this list of possible heads and tails. As a drop in recall can be expected on domain-specific test sets, we use domain-adaptation techniques to combine the large out-of-domain data set (Wikipedia) with the smaller in-domain data sets.

Dutch compound splitter

To ensure a broad coverage of topics, we compiled a frequency list of token-PoS-tag-tuples for Dutch derived from a part-of-speech tagged Dutch Wikipedia dump of 15 million words. We used a coarse-grained PoS tag set of 10 categories that are relevant for compound splitting: plural noun, singular noun, adjective, numeral, adverb, preposition, past participle, present participle, infinitive and verb stem.

We followed the implementation of Réveil and Martens (2008) and stored all possible heads and tails (together with the frequency and PoS information) in two prefix trees. Possible heads or tails are defined as words of minimally three characters, containing at least one vowel. Heads belong to one of the abovementioned PoS categories; tails belong to the same set without adverbs, prepositions and numerals.

As the Wikipedia files were automatically parsed, tokenized and PoS-tagged, they inevitably contain errors. To avoid the problem of error percolation, a minimum frequency threshold was experimentally set at 20. Unfortunately, the frequency threshold could not fully prevent non-words being stored in the prefix trees. Therefore, non-words due to spelling mistakes, (e.g. *vor* instead of *voor* (En: *for*)) or tokenization problems (e.g. *ste* or *ata*) were manually filtered out on the basis of tests on the development corpus (see section Data Sets and Experiments). We also compiled a list of non-productive prepositions and adverbs (e.g. *hoe, dan* and *per* (En: *how, then, per*)) and a list of frequent Dutch derivational suffixes on the basis of the ANS[1], an authoritative Dutch grammar book, which are discarded as possible heads or tails. With the minimum frequency threshold of 20 and after applying the filters described above, the head prefix tree contains 71,147 possible heads and the tail prefix tree 70,189 possible tails.

The compound splitter searches the head and tail prefix trees for all possible split points. The compound splitter allows a linking-s between the head and the tail as is the case in e.g. *aanwezigheid+s+lijst* (En: *attendance list*).

Head and tail combinations are considered valid if the PoS combination is included in a predefined list of valid PoS combinations. This list was compiled on the basis of the development set:

- Noun tails can be combined with nouns, adjectives, adverbs and verb stems as heads;
- Adjective tails can be combined with singular nouns, prepositions, adverbs, adjectives and verb stems as heads;
- Infinitive tails can be combined with prepositions, adverbs, adjectives and past participles;
- Past and present participles as tail can be combined with

[1] http://ans.ruhosting.nl/e-ans/

prepositions, adverbs and adjectives as head.

Two other restrictions limit the number of possible split points. A first restriction blocks two identical consonants followed by the ending –en. This rule prevents the erroneous splitting of Dutch plural forms as in e.g. *boodschappen* (En: *groceries*) into *boodschap* (En: *message*) and *pen* (En: *pen*). The second restriction regulates the linking-s. The linking-s is not allowed if the head is a preposition, adverb or adjective and can only be split off in certain contexts. A list of 2,585 possible contexts (defined as two letters at the left and two letters at the right) was compiled on the basis of a set of 50,000 Dutch compounds (see section Data Sets and Experiments).

The compound splitter generates all possible split points and retrieves the frequency information of the token-PoS-tag-tuples from the suffix trees, after which the split with the highest geometric mean of word frequencies of its parts (Koehn & Knight, 2003) is chosen as the best solution:

$$(\prod_{i=1}^{n} freq_p)^{1/n}$$

, in which *n* is the number of split points in the compound and *freq*$_p$ is the frequency of the component parts.

The following example shows the possible split points for the word *staatsbankroet* and the geometric mean calculated for the different splits:

staat (51657)+*s*+*bankroet* (257)   3643.60   En: *state* +*bankruptcy*

staats (146)+*bankroet* (257)	193.70	En: *of the state +bankruptcy*
staatsbank (24)+*roet* (328)	88.72	En: *state-owned bank+soot*

Note that original tokens, without split points are also considered, as is the case in *databank*:

data (2535)+*bank* (4226)	3273.06	En: *data +base*
databank (224)	224.0	En: *database*

Compounds can be nested and especially in technical texts, compounds of more than two components frequently occur, as is the case in e.g. *satelliet+[navigatie+systeem]* (En: *satellite navigation system*) and *[[baar+moeder]+hals]+kanker* (En: *cancer of the cervix uteri*). Therefore, the compound splitter can further split the component parts in their underlying parts.

Domain adaptation

As mentioned above, we compiled a frequency list on the basis of Wikipedia to ensure a broad coverage of topics. The Wikipedia frequency list is static and forms the core part of the compound splitter. However, as we aim to integrate the compound splitter in a terminology extraction system, we foresee a mechanism to extend the large static Wikipedia frequency list with a smaller dynamically compiled frequency list derived from the extraction corpus. To account for differences in corpus size, the in-domain frequencies are estimated on the basis of their relative frequencies.

Data Sets and Experiments

We compiled three different Gold Standard data sets on the basis of Celex (Baayen, Piepenbrock, & van Rijn, 1993): a set of 50,000 Dutch compounds, a set of 5,000 monomorphemic Dutch words and a development set of 5,550 compounds and 2,886 monomorphemic words. To evaluate the performance of the compound splitter on a more technical domain, we used a set of 5,000 Dutch compounds belonging to the automotive domain that had been compiled for earlier research (Lefever, Macken, & Hoste, 2009) and an in-domain frequency list derived from an automotive corpus of 2.7 million words.

To evaluate the compound splitter, we compare its output with the Gold Standard data and used precision, recall and accuracy as evaluation metrics. These metrics are commonly used in the field (Fritzinger & Fraser, 2010; Koehn & Knight, 2003; Parra Escartín, 2014) and can be defined as follows:

$$Precision = \frac{\#CorrectlySplit}{\#WordsSplit}$$

$$Recall = \frac{\#CorrectlySplit}{\#CompoundsInCorpus}$$

$$Accuracy = \frac{\#CorrectWords}{\#WordsInCorpus}$$

We experimented with different minimum frequency thresholds and we also defined a minimum length threshold (expressed in the number of characters) for words to be sent to the compound splitter. As expected, raising the minimum frequency threshold and the minimum length threshold has a positive impact on precision, but lowers recall. The results reported in Table 1 use a minimum frequency threshold of 20 and a minimum length threshold of 7 characters.

On the test set of 5,000 monomorphemic words, the compound splitter reaches an accuracy of 98.3. It wrongly split 84 monomorphemic words of which 60 are Dutch infinitives such as *mopperen* (En: *grumble*), which is wrongly split in *mop+peren* (En: *joke + pears*).

On the Celex test set of 50,000 Dutch compounds the compound splitter has a precision of 98.5 and a recall of 80.3 if the words are split at the highest level (1-level compound splitting). These figures drop tot 94.9 and 77.4 in the case of 2-level compound splitting.

Please note that we adopt a very strict evaluation method. If we ignore the linking-s and append it to the head in both the Gold Standard data set and the output of the compound splitter (as in *varken+s+snuit* → *varkens+snuit*, En: *pig's snout*), this operation solves 47% of the wrongly split words.

Precision and recall scores on the test set consisting of 5,000 compounds of the automotive domain are slightly lower (a precision score of 97.8 and recall score of 76.6 for 1-level compound splitting and a precision score of 88.6 and recall score of 69.9 for 2-level compound splitting). The lower 2-

level scores for the automotive test set can be attributed to the higher percentage of nested compounds in the technical data set (22.8% vs. 5.8% in the Celex data set).

We also tested the compound splitter using the in-domain frequency list of the automotive corpus of 2.7 million words instead of the Wikipedia frequency list. Precision and recall scores are slightly lower for 1-level compound splitting and remarkably lower for 2-level compound splitting. These figures demonstrate that the Wikipedia frequency list indeed has a good coverage of more technical domains. By combining both frequency lists, best recall scores are obtained while the precision scores only slightly decrease.

Test corpus and frequency information used	**Precision**	**Recall**
1-level compound splitting		
Celex (Wikipedia freq. list)	98.5	80.3
Automotive (Wikipedia freq. list)	97.8	76.6
Automotive (Automotive freq. list)	97.2	75.7
Automotive (Wikipedia and automotive freq. list)	96.4	88.5
2-level compound splitting		
Celex (Wikipedia freq. list)	94.9	77.4
Automotive (Wikipedia freq. list)	88.6	69.4
Automotive (Automotive freq. list)	83.9	65.3
Automotive (Wikipedia and automotive freq. list)	86.5	79.4

Table 1: precision and recall scores on two test sets using different frequency lists

In a real terminology extraction scenario however, much smaller extraction corpora are available. We therefore created two smaller parallel data sets to test the compound splitter and the impact on word alignment and subsequent terminology extraction.

The first one is an English-Dutch corpus belonging to the medical domain, and consists of four European public assessment reports (EPARs) extracted from the Dutch Parallel Corpus (Macken, De Clercq, & Paulussen, 2011). It is a relatively small corpus and contains 4,333 English and 4,332 Dutch tokens. Manual word alignments are available for this data set in the Dutch Parallel Corpus.

The second corpus is a French-Dutch parallel corpus belonging to the automotive domain of 14,087 French and 13,133 Dutch tokens. It is a subset of the data set used in (Lefever et al., 2009) for which manual word alignments are also available.

Again, we contrast the performance of the compound splitter using the Wikipedia frequency list with one using a combined version of the Wikipedia frequency list and a frequency list derived from the Dutch part of the in-domain parallel corpus. Despite the fact that the in-domain frequency lists are much smaller than in our previous experiments, using additional in-domain data drastically increases precision and recall scores for the medical domain and increases the recall scores in the automotive domain.

We PoS-tagged the parallel corpora and evaluated the performance of the compound splitter only on nouns and adjectives. The basic underlying assumption is that especially nouns and adjectives are important for terminology extraction. Limiting compound splitting only to nouns and adjectives yields the best overall results.

1-level compound splitting	**Precision**	**Recall**	**Accuracy**
Wikipedia freq. list	72.7	56.2	96.3
Wikipedia and medical freq. list	77.7	74.9	97.3
Wikipedia and medical freq. list, restricted to nouns and adjectives	80.5	74.1	**98.0**
2-level compound splitting			
Wikipedia freq. list	72.2	55.8	96.3
Wikipedia and medical freq. list	76.9	74.1	97.3
Wikipedia and medical freq. list, restricted to nouns and adjectives	79.5	73.1	**98.0**

Table 2: precision and recall scores on the medical data set in different settings

Analysing the output of the compound splitter on the medical data set we see that the splitter misses compounds such as *injectie+flacons* (En: *vials*) whose compound parts are not present in the Wikipedia frequency list and do not occur as a single word in the Dutch part of the small parallel data set.

Wrongly split compounds are monomorphic words such as *receptoren* (En: *receptors*), which was erroneously split in *recept+oren* (En: *recipe* + *ears*) or *besloot* (En: *concluded*), which was erroneously split in *bes+loot*. Limiting compound splitting only to nouns and adjectives solved the last case.

1-level compound splitting	**Precision**	**Recall**	**Accuracy**
Wikipedia freq. list	88.9	62.1	93.6
Wikipedia and automotive freq. list	88.8	66.9	94.2
Wikipedia and automotive freq. list, restricted to nouns and adjectives	90.1	65.9	**94.7**
2-level compound splitting			
Wikipedia freq. list	81.8	57.1	92.9
Wikipedia and automotive freq. list	79.8	59.9	93.2
Wikipedia and automotive freq. list, restricted to nouns and adjectives	80.0	58.4	93.8

Table 3: precision and recall scores on the automotive data set in different settings

A manual inspection of the missed compounds reveals a phenomenon that frequently occurred in the automotive data set and that cannot be handled by the current system. The head of compounds such as *opberg+vak* (En: *stowage box*) or *aandrijf+tandwiel* (En: *drive pinion*) consists of a verb form that never occurs as such in a corpus as the prefix of a separable verb is separated from the verb (*berg…op, drijf…aan*). Only in the infinitive and the past participle the separable verb is written as one word (*opbergen, aandrijven*). Rules for such type of transformations are currently lacking in the system.

At this moment, the compound splitter does not use the PoS code of the compound. However, as the compound inherits the PoS category of the tail, putting a restriction on the tail's PoS category to match the compound's PoS category will avoid errors such as *stekkers* (En. *plugs*), which is wrongly split in *stek+kers* (En: *spot + cherry*) and the case of *mop+peren* described above.

Impact on word alignment

In statistical machine translation, translational correspondences are estimated from bilingual corpora on the basis of statistical word alignment models that are based on the assumption of co-occurrence: words that are translations of each other co-occur more often in aligned sentence pairs than that they occur randomly.

In the context of statistical machine translation, GIZA++ is one of the most widely used word alignment toolkit. GIZA++ implements the IBM models 1–5 (Brown et al., 1993) and is used in Moses (Koehn et al., 2007), an open-source statistical machine translation system.

One of the shortcomings of the IBM models is that they only allow one-to-many word mappings as they take the source word as their starting point to estimate conditional probabilities (i.e. the probability that a target word is a translation of a source word, given the source word). Multiword units (e.g. the Dutch word *regeringsleider* (En: *Head of Government*) are problematic for the alignment models, as every word (*Head*, *of* and *Government*) is treated as a separate entry. To overcome this problem, the IBM models are used in two directions: from source to target and from target to source after which a symmetrization heuristic (Koehn et al., 2005) combines the alignments of both translation directions. *Intersecting* the two alignments results in an overall alignment with a higher precision, while taking the *union* of the alignments results in an overall alignment with a higher recall. The default symmetrization heuristic applied in Moses (*grow-diag-final*) starts from the intersection points and gradually adds alignment points of the union to link unaligned words that neighbor established alignment points.

The main problem with the *union* and the *grow-diag-final* heuristics is that the gain in recall causes a substantial loss in precision, which poses a problem for applications such as terminology extraction in which precision is important.

Apart from the one-to-many word alignment problem, compounds also lead to data sparseness. The compounding process is highly productive and can create a potentially infinitive number of valid Dutch words, which as a consequence occur infrequently in the data sets that are used to train the word alignment models. As terminology extraction systems typically work with much smaller corpora than machine translation system, this makes the problem of data sparseness even more apparent.

A solution to overcome the problems of data sparseness and the one-to-many alignments is to split compounds into their component parts prior to word alignment (Koehn & Knight, 2003; Stymne & Holmqvist, 2008).

However, the underlying assumption that compounds are translated compositionally is not always valid. The assumption holds for examples such as *injectie+oplossing* (En: *solution for injection*), but not for *doktervoorschrift* (En: *prescription*) or *werkbank* (Fr: *établi*, En: *workbench*)

This observation led us to investigate a new approach in which we train the word alignment models twice: a first time on the original data set and a second time on the data set in which the compounds are split into their component parts. We then apply the normal *intersection* heuristics on both data sets after which we merge all alignment points. We then expand this model by adding alignment points from the *grow-diag-final* output of the word alignment model trained on the split compounds data set. An alignment point is added to the merged alignments if the following conditions are met:

- The source alignment point is new (source language is the language which is not subject to compound splitting in our experiments)
- The target alignment point is a compound

Data sets and Experiments

To evaluate the impact of compound splitting (1- and 2-level splitting) and the different word alignment scenarios we used the two terminology extraction corpora described above for which we have manual word alignment available.

To evaluate the system's performance, we used the evaluation methodology of Och and Ney (2003), who introduced the following redefined precision and recall measures,

$$precision = \frac{|A \cap P|}{|A|},\ recall = \frac{|A \cap S|}{|S|}$$

and the alignment error rate:

$$AER(S,P,A) = 1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}$$

in which **S** refers to sure alignments, **P** to possible alignments (which also includes the sure alignments) and **A** to the set of alignments generated by the system.

We built different word alignment systems and compared systems with no compound splitting (NC) with manual compound splitting (MC), level 1 (L1) and level 2 (L2) and automatic compound splitting (AC), level 1 and level 2.

As a first experiment we use the methodology that is commonly used in machine translation and split compounds into their component parts prior to word alignment after which we apply the different symmetrization heuristics (intersection, union and grow-diag-final) provided in Moses. To avoid error percolation, we work with the manually split compounds.

In table 4 below we see that the best precision, recall and AER scores are all obtained with the word alignment models after compound splitting, for both data sets, from which we can conclude that high-quality compound splitting improves word alignment quality.

	Medical En-Nl			**Automotive Fr-Nl**		
Setting	**Prec.**	**Rec.**	**AER**	**Prec.**	**Rec.**	**AER**
NC intersect	93.69	68.52	20.58	93.77	55.65	30.15
NC gdf	75.00	82.02	21.78	76.43	76.08	23.73
NC union	71.22	84.29	23.07	73.81	78.13	24.90
MC L1 intersect	93.91	70.36	**19.03**	**95.09**	58.74	27.37
MC L1 gdf	74.95	82.72	21.52	76.84	78.33	22.41
MC L1 union	70.78	**84.58**	23.24	74.30	79.65	23.21
MC L2 intersect	**93.96**	70.05	19.18	94.82	58.88	27.34
MC L2 gdf	74.50	82.43	21.89	77.41	78.98	**21.82**
MC L2 union	70.68	84.39	23.37	75.09	**80.32**	22.37

Table 4: precision, recall and AER on the medical and automotive data set trained on the original data (NC) and the data set in which compounds are split manually (MC)

Next we merge (MRG) all high-quality alignment points obtained by the intersection heuristic on both data sets (NC intersect and MC L1 intersect or MC L2 intersect). As can be seen in table 5, merging the two sets of intersected alignment points improves recall and AER scores for both data sets compared to the intersection on the original data set (NC intersect in table 4) or the intersection on the data set in which the compounds are split

(MC L1 intersect of MC L2 intersect in table 4). For the automotive data set, using a second level of compound splitting further improves the recall scores.

	Medical En-Nl			Automotive Fr-Nl		
Setting	**Prec.**	**Rec.**	**AER**	**Prec.**	**Rec.**	**AER**
MRG MC L1-NC	92.17	73.42	**18.01**	92.53	62.45	25.41
MRG MC L2-NC	93.96	70.05	19.18	92.25	63.01	**25.12**

Table 5: precision, recall and AER on the merged intersected alignment points (medical and automotive data set)

Finally, as still a lot of alignment points are missing in the data, we add additional alignment points taken from the *grow-diag-final* set trained on the split compounds corpus provided that they meet the requirements explained above (new alignment point for source word, target word is a compound). Adding these additional alignment points improves the recall scores for both data sets further, while marginally reducing precision.

	Medical En-Nl			Automotive Fr-Nl		
Setting	**Prec.**	**Rec.**	**AER**	**Prec.**	**Rec.**	**AER**
MRG+GDF MC L1-NC	89.46	74.49	18.50	88.81	68.92	22.38
MRG+GDF MC L2-NC	90.23	74.97	**17.89**	89.28	68.78	**22.29**

Table 6: precision, recall and AER on the merged intersected alignment points enriched with alignment points taken from the grow-diag-final set trained on data sets in which the compounds were manually split (medical and automotive data set)

The results in tables 4, 5 and 6 demonstrate that high-quality (manual) compound splitting improves word alignment quality. We now repeat the experiments by using the automatically split compounds. Table 7 presents the results of the merged intersected alignment points (original data and automatically split data) enriched with alignment points taken from the grow-diag-final set of the automatically split data. On the medical data set, the obtained scores for 1-level splitting approximate the scores of the manual compound splitting, while 2-level splitting seems to work best for the automotive data set.

The lower part of the table presents the results when limiting compound splitting only to nouns and verbs. This has a minor positive impact on the medical data set. As the automotive data set contains really technical texts, the PoS tagger probably introduces too many errors to be fully reliable.

	Medical En-Nl			**Automotive Fr-Nl**		
Setting	**Prec.**	**Rec.**	**AER**	**Prec.**	**Rec.**	**AER**
MRG+GDF AC L1-NC	90.46	73.74	**18.52**	90.13	66.45	23.49
MRG+GDF AC L2-NC	90.08	73.14	19.08	90.11	66.60	**23.40**
MRG+GDF AC L1-NC F	90.65	72.87	18.98	90.34	66.29	23.49
MRG+GDF AC L2-NC F	90.88	72.89	18.86	90.60	66.32	23.40

Table 7: precision, recall and AER on the merged intersected alignment points enriched with alignment points taken from the grow-diag-final set trained on data sets in which the compounds were automatically split, without and with PoS filtering (medical and automotive data set)

From the experiments with the automatically split compounds we can conclude that even with imperfect compound splitting high-precision word alignment can be obtained with reasonable recall scores, even when trained on small parallel data sets.

Impact on terminology extraction

We evaluated the different word alignment scenarios in the TExSIS terminology extraction system, which is a more advanced system of the system described in Macken, Lefever, and Hoste (2013). The TExSIS system is a hybrid system that uses both linguistic and statistical information. The bilingual terminology extraction system first generates monolingual term lists for the source and target part of the extraction corpus, after which source and target terms are paired on the basis of word alignments.

The monolingual term extraction component produces a list of term candidates on the basis of predefined morpho-syntactic patterns. Two statistical filters are then used to create the final term list: Log-Likelihood ratio is applied on all single-word terms to filter out general vocabulary words; C-value (Frantzi & Ananiadou, 1999) is calculated for all multi-word terms to determine unithood (Kageura & Umino, 1996).

The bilingual term extraction component uses the word alignments to pair source and target terms. Term pairs are valid if for all source and target content words alignment points are found within the term pair and if there are no alignments points from words within the term pair to words outside the term pair. As such, the success of the term pairing process heavily depends on the quality of the word alignments. A high precision is extremely important to pair single word terms, whereas a high recall is also important to pair multi-word terms.

The TExSIS terminology extraction system integrates the word alignments of Moses described above. We created three baseline systems with TExSIS (without the compound splitter) using the three different symmetrization heuristics, viz. intersection, grow-diag-final, and union.

Compounds are problematic in this framework as they are often erroneously paired with a partial translation due to missing word alignments. A typical example is the erroneous term pair *dose* - *aanvangsdosis*, which should be paired *starting dose* - *aanvangsdosis.*

Experiments

To evaluate the impact of compound splitting on bilingual terminology extraction we created Gold Standard bilingual term lists for the two domain-specific parallel corpora described above. As the aim of the Gold Standard term lists is to test the impact of compound splitting on the bilingual term extraction module, the term lists only contain valid term pairs, so source or target terms for which no valid counterpart is found in the translation are discarded. The English-Dutch medical bilingual term list contains 369 term pairs of which 96 Dutch paired terms contains a compound (26%) and the French-Dutch automotive term list contains 1,909 term pairs of which 1,109 Dutch paired terms contains a compound (58%).

We evaluated different word alignment scenarios in the TExSIS bilingual term extraction module and report precision, recall and the harmonic mean F. The results are presented in table 8. As upper bound we used the manually created word alignments described above. On the medical data set, the upper bound precision score is 54.76 and the upper bound recall score is 57.72. Precision scores are higher on the automotive data set (66.93), but recall scores are lower (45.15).

The upper bound figures demonstrate that (bilingual) terminology extraction is a difficult task. A manual inspection of the wrong and missed term pairs using the manual word alignments shows us that most wrong terms pairs are terms that are not specific enough such as the term pair *ingredient* - *stof* or are larger multiword terms that are not part of the Gold Standard data set as such, but whose parts are included in the Gold Standard data set, e.g. *masse du piston* - *massa van de zuiger* (the smaller parts *masse – massa* and *piston* - *zuiger* are included in the Gold Standard). Missed terms pairs are o.a. adjectives and verbs that are currently not extracted, e.g. *spread* - *uitzaaien* and *unresectable* - *niet-operabel*.

	Medical En-Nl			**Automotive Fr-Nl**		
Setting	**Prec.**	**Rec.**	**F**	**Prec.**	**Rec.**	**F**
Manual word alignments	54.76	57.72	**56.20**	66.93	45.15	**53.93**
NC intersect	48.52	48.78	**48.65**	47.78	29.91	36.79
NC gdf	51.50	42.01	46.27	61.59	32.84	**42.84**
NC union	53.23	37.94	44.30	65.62	30.59	41.73

MRG+GDF MC L1-NC	53.49	53.93	**53.71**	63.70	38.24	47.79
MRG+GDF MC L2-NC	52.80	53.66	53.23	65.23	38.82	**48.67**
MRG+GDF AC L1-NC	50.93	52.03	51.47	59.31	36.72	45.36
MRG+GDF AC L2-NC	51.19	52.57	**51.87**	59.93	36.83	45.62
MRG+GDF AC L1-NC F	51.77	51.49	51.63	59.46	37.04	45.64
MRG+GDF AC L2-NC F	50.80	51.76	51.28	59.75	37.24	**45.89**

Table 8: Term extraction results: precision, recall and F-score using different word alignment scenarios (medical and automotive data set)

If we look at the results of the standard TExSIS system without compound splitting (NC intersect, NC gdf and NC union), we observe a different behaviour on the two data sets. Intersection (NC intersect) yields the best results on the medical data set but the worst on the automotive data set. Substantial improvements can be achieved by using the proposed word alignment technique described above on the data set containing the manually split compounds (MRG+GDF MC L1/L2-NC). Two-level compound splitting gives the best overall results on the automotive data set. Automatic compound splitting also improves the results considerably. On both data sets best results are obtained using two-level compound splitting. Filtering on PoS code only leads to a minor improvement on the automotive data set.

Conclusion

We described a compound splitting method for Dutch, which uses frequency information and linguistic knowledge to determine the split points. To optimize the performance of the compound splitter on domain-specific data sets, we combine a dynamically compiled in-domain frequency list with the large static Wikipedia frequency list. To account for nested compounds, the compound splitter can generate compounds at different levels. We experimented with 1- and 2-level splitting.

We developed a novel methodology to incorporate compound splitting in word alignment. Rather than choosing for data sets with or without split compounds, we train the word alignment models twice: a first time on the original data set and a second time on the data set in which the compounds are split into their component parts. We merge the intersected alignment sets to obtain high precision alignment points which are then further enriched by adding selected alignment points from the grow-diag-final set of the split compounds corpus.

The obtained (precise) word alignments are integrated in the TExSIS bilingual terminology extraction system. The novel word alignment technique substantially improves terminology extraction results if manually split compounds are used and considerably improves the results when the compounds are split automatically.

As the compound splitting tool can still be improved by implementing a PoS restriction so that the PoS code of the tail matches the PoS code of the compound and by allowing morphological operations on separable verb forms, we are confident that the results on terminology extraction can still be improved.

For machine translation purposes 1-level compound splitting is considered to be sufficient (Fritzinger & Fraser, 2010). In our experiments, 2-level compound splitting led to the best results. In future work, we will implement a recursive call in the system and experiment with all possible levels. We will also evaluate whether machine translation also benefits from our novel word alignment method.

References

Baayen, R. H., R. Piepenbrock, & H. van Rijn. (1993). The CELEX lexical database on CD-ROM. Philadelphia, PA: Linguistic Data Consortium.

Brown, P. F., V. J. Della Pietra, S. A. Della Pietra, & R. L. Mercer. 1993. "The Mathematics of Statistical Machine Translation: Parameter Estimation". *Computational Linguistics, 19*(2), 263-311.

Frantzi, K., & S. Ananiadou. 1999. "The C-value / NC-value domain independent method for multi-word term extraction". *Journal of Natural Language Processing, 6*(3), 145-179.

Fritzinger, F., & A. Fraser. 2010. "How to avoid burning ducks: combining linguistic analysis and corpus statistics for German compound processing". In *Proceedings of the ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. 224-234. Uppsala, Sweden.

Kageura, K., & B. Umino. 1996. "Methods of automatic term recognition. A review". *Terminology, 3*(2), 259-289.

Koehn, P., A. Axelrod, A. Birch Mayne, C. Callison-Burch, M. Osborne, & D. Talbot. 2005. "Edinburgh system description for the 2005 IWSLT speech translation evaluation". In *Proceedings of the International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation (IWSLT 2005)*. Pittsburgh, PA, USA.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, et al. 2007. "Moses: Open Source Toolkit for Statistical

Machine Translation". In *Proceedings of the ACL 2007 Demo and Poster Sessions*. 177-180. Prague, Czech Republic.

Koehn, P., & K. Knight. 2003. "Empirical methods for compound splitting". In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*. 187-193. Budapest, Hungary.

Lefever, E., L. Macken, & V. Hoste. 2009. "Language-independent bilingual terminology extraction from a multilingual parallel corpus". In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. 496-504. Athens, Greece.

Macken, L., O. De Clercq, & H. Paulussen. 2011. "Dutch Parallel Corpus: a Balanced Copyright-Cleared Parallel Corpus". *Meta, 56*(2), 374-390.

Macken, L., E. Lefever, & V. Hoste. 2013. "TExSIS. Bilingual terminology extraction from parallel corpora using chunk-based alignment". *Terminology, 19*(1), 1-30.

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

Parra Escartín, C. 2014. "Chasing the Perfect Splitter: A Comparison of Different Compound Splitting Tools". In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. 3340-3347. Reykjavik, Iceland.

Réveil, B., & J.-P. Martens. 2008. "Reducing speech recognition time and memory use by means of compound (de-)composition". In *Proceedings of the Annual Workshop on Circuits, Systems and Signal Processing (ProRISC 2008)*. 348–352. Utrecht, The Netherlands.

Stymne, S., & M. Holmqvist. 2008. "Processing of Swedish compounds for phrase-based statistical machine translation". In *Proceedings of the 12th annual conference of the European Association for Machine Translation (EAMT 2008)*. 182-191. Hamburg, Germany.