

---

# Evaluating Compound-to-compound Links in a Sub-sentence Aligned Bilingual Corpus through Example-based Element Recognition

MARKUS SAERS

Department of Linguistics and Philology, Uppsala University  
markuss@stp.ling.uu.se

**ABSTRACT.** This paper will present an algorithm that evaluates links between one-word compounds and two-word compounds in a bilingual corpus that has been aligned at the sub-sentence level. The phenomenon of linking one-word compounds to multi-word compounds is common when English is being linked to other Germanic languages, and it is difficult to get the links right in the alignment process. The algorithm has been tested on a Swedish–English bilingual corpus with encouraging results.

## 1 Introduction

In this paper I will describe a novel approach to evaluating links between Swedish one-word compounds and English free-form compounds. The basic idea is that the link is valid if the Swedish compound can be segmented according to an example set by the English compound. In order to achieve this, a novel approach to example based compound segmentation in this particular context is presented.

When automatically aligning the English side of an English–Swedish bilingual corpus with the Swedish side, a large part of the errors occur because English and Swedish use different methods to compound words. Where English juxtaposes elements to form multi-word compounds, Swedish (and other Germanic languages) concatenates them to form one-word compounds. This means that there is no one-to-one correspondence between words on the two sides of the corpus. There are two ways around this problem: aligning several English words to one Swedish word, or aligning one English word to a Swedish sub-word unit. The most common approach seems to be the first one (see e.g. Dagan et al. 1993, Melamed 1997 and Och and Ney 2000), with the notable exception of Koehn and Knight (2003). The problem with the first approach is to find the boundaries of the English multi-word unit, this causes between 2.5 and 5.6 % of the errors reported in Ahrenberg

et al. (1998), who specifically report numbers on partial linkage. The algorithm in this paper aims at weeding out these partial links from an automatically generated linking material.

Needless to say, a faulty lexicon will affect the system it is used in, and a faulty bilingual lexicon can have devastating effects on any machine translation system.

Please note that this paper reports findings that are more elaborately described in my Master's thesis (Saers 2005).

## 2 Identifying candidate compounds

In this paper, only morphemes that have a meaning of their own, and have a root that can act as a word, will be considered compound elements. This means that the word *lejonunge* 'lion cub' will be considered a compound since both *lejon* 'lion' and *unge* 'cub' can appear as separate words, whereas *lejoninna* 'female lion' will not be considered a compound, since *inna* is a morpheme used to derive feminine form, and cannot be used as a word on its own.

Also, a very crude method will be applied to identify candidate compounds in the material. As the material consists of links between Swedish and English link units, a candidate compound will be defined as a link connecting a Swedish one-word unit to an English multi-word unit. All links fitting this description will be considered candidate compounds.

## 3 Extracting sub-word strings – silhouettes

In this section a method for segmenting one-word compounds is presented, which is imperative to this approach. It is loosely based on the Longest Common Sub Sequence (LCSS), a measurement of how similar two strings are, used mainly to identify cognates (see e.g. Simard et al. 1992). Cognates are words that look very orthographically similar when translated into another language. An example is the word *example*, which translates into *exempel* in Swedish. The LCSS between the two words would be *exmpe* or *exmpl*, in either case five letters out of seven. When comparing the LCSS of two concatenated compounds containing the same element, it is possible to extract a string representation of that element in the process. Creating silhouettes is a way of doing this.

To extract elements from compounds, two compounds, that both contain the same element, are needed. All sub strings shared between the two compounds are potential elements, and can be extracted by creating a table similar to that created when calculating the LCSS. Instead of filling the cells with the LCSS so far however, the cells are filled with ones and zeros, denoting a match and a mismatch respectively. Consecutive, diagonal ones now denote extractable strings. Figure 23.1 shows such a table, with the extractable strings encircled.

A silhouette does not only represent a sub string, but also where in the original string it

	t	r	y	c	k	m	ä	t	a	r	e
t	1	0	0	0	0	0	0	1	0	0	0
r	0	1	0	0	0	0	0	0	0	1	0
y	0	0	1	0	0	0	0	0	0	0	0
c	0	0	0	1	0	0	0	0	0	0	0
k	0	0	0	0	1	0	0	0	0	0	0
v	0	0	0	0	0	0	0	0	0	0	0
a	0	0	0	0	0	0	0	0	1	0	0
k	0	0	0	0	1	0	0	0	0	0	0
t	1	0	0	0	0	0	0	1	0	0	0

Figure 23.1: The table produced when extracting sub strings. Encircled areas represent the extracted substrings.

was found. Therefore silhouettes must know how the original string looks. This relationship is expressed in text as a silhouette *over* a word. To know where in the string the sub string was found, a silhouette is constructed to be of the same length as the original string, and consist of ones and zeros in the same way as they are distributed in the table. Figure 23.1 thus contains seven silhouettes over the two words (two which are identical). For the word *tryckvakt*, the silhouettes would be a set like this: {111110000, 100000000, 010000000, 000000100, 000000010, 000000001}. The second silhouette, denoting the latter ‘t’ is found twice in the table, but since a set is created, all identical silhouettes are collapsed into one.

To facilitate the reading of silhouettes, a one may be replaced by the corresponding character, and a zero with a ‘•’. The above set would then look like this: { tryck••••, t••••••••, ••••••••, ••••••••a••, ••••••••k•, ••••••••t }. All of these six silhouettes denote a possible element that is shared between the two compounds. The actual shared element (*tryck*) is represented by the first silhouette: tryck••••.

## 4 Combining silhouettes

The silhouettes extracted in the previous section can be combined to model a whole compound. To illustrate, assume there is a link looking like this:

tryckvakt : pressure monitor

Using the above method to extract silhouettes shared between *tryckvakt* ‘pressure monitor’ and other Swedish compounds which are linked to either *pressure* or *monitor*, silhouettes can be obtained, which should correspond to the elements that translate into *pressure* and *monitor* respectively. A compound linked to *pressure* might be *tryckmätare* ‘pressure gauge,’ while one linked to *monitor* might be *ångvakt* ‘steam monitor’. The set of silhouettes shared between *tryckvakt* and *tryckmätare* should contain the silhouette tryck••••, while the set shared between *tryckvakt* and *ångvakt* should contain •••••vakt. There will

probably be a lot more silhouettes, but in this example the focus will be on the two correct ones. To combine these two silhouettes, exclusive-or is used. This is the notion of “a or b but not both.” As seen in the example below, the two silhouettes combine to form the original compound:

```
111110000 tryck...
000001111 .....vakt
===== ===== XOR
111111111 tryckvakt
```

It is important to use exclusive-or, since overlapping is unwanted. Overlapping segments get cancelled out by the exclusiveness of the operator. However, not all compounds consists of two elements, so a multi-argument exclusive-or operator is needed. This would be the notion of “one of  $a_1, a_2, \dots, a_n$ , but only one”.

A silhouette derived from the combination of other silhouettes will be called a complex silhouette below, and it will be assumed that it has access to the silhouettes making it up.

## 5 Evaluating silhouette links

A silhouette link is a link from the automatic alignment process, where the link units on both sides have silhouettes over them. The link “tryckvakt : pressure monitor” can be expanded into this silhouette link:

```
{.....}*   pressure.....
{.....}*   .....monitor
=====   ===== XOR
tryckvakt   pressure.monitor
```

The units marked with an asterisk are sets of silhouettes, all possibly fitting in the spot. The silhouette set corresponding to “pressure” would, for example, be a superset of the example set extracted in section 3. The problem now is to choose the correct silhouettes from these sets. In this section, a method for evaluating silhouettes in this context is presented.

One obvious measurement that should be considered is *frequency*. The more common silhouettes should be preferred over the less common. Frequency has, however, a drawback: the smaller the unit the more common it will be. The letter ‘a’ for example, is bound to be in a lot of words, and will therefore occur more frequently than the correct element. This has to be counterweighted by another measurement, in this case *cover*. Cover is a measurement of how much of the compounds the silhouette covers. The one-character silhouettes are now at a disadvantage to the longer, correct ones.

A third measurement that proved useful in the development process was to compare the relative length of the Swedish candidate elements to the relative length of their corresponding English elements. To do this, the *likeness* measurement was introduced. Likeness is defined in terms of differences between the silhouettes, and is calculated as 1-difference.

The difference between silhouettes is calculated by summing up the relative length differences between the silhouettes on either language side, like this:

tryck....	5/9=55.56%		pressure.....	8/16=50.00%		55.56-50.00=5.56
.....vakt	4/9=44.44%		.....monitor	7/16=43.75%		44.44-43.75=0.69
=====			=====			
tryckvakt	9/9=100.00%		pressure.monitor	15/16=93.75%		100.00-93.54=6.25
						=====
						12.50

The likeness of this silhouette link is thus  $100 - 12.5 = 87.5$  %. To use relative length between two languages is not uncontroversial, but in this case, where the words compared are Swedish and English technical terms, it works sufficiently well to deserve a try.

The final evaluation metric is a weighted mean between these three measurements, called merit. The weights were established through training, and the findings are reported in section 6.

To evaluate links, the silhouette link is evaluated as described above. If the merit value is higher than “keep threshold” the link is kept, and if it is lower than a “discard threshold” it is discarded. Should the merit value fall between the two thresholds, the evaluation is inconclusive.

## 6 Training

Training consists of establishing the weights for the different measures used to calculate merit value (section 6.1), and establishing thresholds for evaluating the merit value (section 6.2).

### 6.1 Training the silhouette evaluation

This section describes how the weights were trained for the different evaluation measurements. The measurements are likeness to linked unit, frequency and link unit coverage. A part of the PLUG-corpus (Sågvald Hein 1999) was used as training material. The part used consists of approximately 100,000 words of technical text (software manuals). The corpus has been aligned at sub-sentence level using UWA (Uppsala Word Aligner, Tiedemann 1999). An a priori reference was created consisting of all candidate compounds (Swedish one-word units linked to English multi-word units), and different settings were tried to see which one recognized most elements correctly. The reference represents the judgement a human would make.

When evaluating the simple silhouettes, likeness is the most important factor, followed by frequency, and last cover.

Also, one has to keep in mind that the training process was carried out on a “contaminated” material, which might have influenced the process. Had the training been carried out on a corrected material the method might have worked better.

## 6.2 Training the link evaluation

The training of the link evaluation aims at establishing merit thresholds for when a link is deemed acceptable, and when it is not. There may be a gap between the thresholds which corresponds to a situation where the algorithm is unable to determine whether a link should be kept or not.

To evaluate different threshold settings, an a posteriori reference was created from a sample of the training material (same as for training silhouette evaluation). A total of 499 links were randomly selected, and a predetermined strategy was used to evaluate every link by hand. It turned out that 390 of the 499 were acceptable and should be kept, while 109 were unwanted and should be discarded. This shows that compounds are especially difficult to align correctly, as there is an accuracy of only 78.16 % in the material containing only links between candidate compounds.

All threshold combinations, where the keep threshold was larger than or equal to the discard threshold, were tested. The thresholds were never set to fractions of percent. In the notation discard threshold/keep threshold, the tested threshold levels were: 0/0, 0/1, 0/2, ..., 0/100, 1/1, 1/2, ..., 1/100, ... 99/100, 100/100. A total of 5,151 different threshold settings were thus tested. Needless to say, the comparison to the reference and calculation of precision and recall worked automatically.

To be able to evaluate which threshold settings yielded the best results, the objective of the process had to be specified. Three different objectives emerged, each potentially needing different threshold settings:

1. To verify part of the material as correct. This means that the focus will be on the links that are deemed acceptable, and there will be an emphasis on precision over recall.
2. To clean out bad links from the material. This means that the focus will be on the links that are deemed unacceptable, and there will be an emphasis on recall over precision.
3. General performance, which can be used to compare the algorithm with future methods. The aim of this objective is to get a good f-score in total.

The settings that succeeded the best with the three objectives are found in table 23.1. These are the settings that will be used in testing. As can be seen in the table, the second objective (second row in the table, cleaning out bad links), does not fare well enough to be considered a viable option. Even so, it will be tested below.

## 7 Testing

The method was tested on two additional corpora that had undergone the same processing as the training corpus. The first corpus was the MATS-corpus (Sågvall Hein et al. 2002). It consists of around 100,000 words of technical text (truck service manuals). The second

Thresholds		Total			Keep			Discard		
discard	keep	prec.	recall	f-score	prec.	recall	f-score	prec.	recall	f-score
n/a	74	n/a	n/a	n/a	94.01	68.46	79.23	n/a	n/a	n/a
30	n/a	n/a	n/a	n/a	n/a	n/a	n/a	57.14	29.36	38.79
27	64	88.71	67.74	76.84	92.49	78.97	85.20	62.50	27.52	38.22

Table 23.1: Summary of the best threshold settings to achieve the three different objectives (one on each row in numeric order). All numbers are in percent. The values marked as “n/a” are of no interest, and may vary.

Thresholds		Total			Keep			Discard		
discard	keep	prec.	recall	f-score	prec.	recall	f-score	prec.	recall	f-score
n/a	74	96.47	54.60	69.73	96.47	60.40	74.29	n/a	n/a	n/a
30	n/a	n/a	n/a	n/a	n/a	n/a	n/a	23.08	25.00	24.00
27	64	87.72	70.00	77.86	95.77	75.22	84.26	22.73	20.83	21.74

Table 23.2: Test results for the three different objectives on technical text. All numbers are in percent. Values set to “n/a” are of no interest to the respective objectives.

corpus was another part of the PLUG-corpus (Sågvalld Hein 1999); this part consisting of about 132,000 words of literary text.

It is important to distinguish between text genres, as technical text contains far more candidate compounds than literary text. When aligned, 17.6 % of the type links in the technical text was candidate compounds, while only 5.6 % of the type links in the literary text. This fact would suggest that the method presented in this paper would do better on compound-rich, technical text than on literary text.

The results from test runs using the settings arrived at above on the two test corpora can be seen in table 23.2 (technical text) and table 23.3 (literary text). As can be seen, the second objective (cleaning out bad links) works as bad in test as in training, rendering it unattainable. The first objective (verifying a part of the material) works approximately as well as in training. Precision is higher but recall is lower. The literary text differs so much that one has to ask the question whether the settings arrived at by the training process are suitable to the literary genre. Perhaps the method needs to be trained specifically towards a genre.

The third objective (general performance) shows a result that might be interesting to use in order to attain the first objective. This depends on how much precision you are willing to trade in for recall, and such a discussion is best taken when a real application is developed. The thresholds presented here were chosen to meet hypothetical goals, not real ones.

Thresholds		Total			Keep			Discard		
discard	keep	prec.	recall	f-score	prec.	recall	f-score	prec.	recall	f-score
n/a	74	97.33	32.44	48.67	97.33	37.73	54.38	n/a	n/a	n/a
30	n/a	n/a	n/a	n/a	n/a	n/a	n/a	18.82	55.56	28.11
27	64	57.06	45.78	50.80	95.58	44.70	60.92	18.33	52.38	27.16

Table 23.3: Test results for the three different objectives on literary text. All numbers are in percent. Values set to “n/a” are of no interest to the respective objectives.

## 8 Conclusions

The conclusion drawn from this experiment is that it is possible to segment one-word compounds according to an example set by a translation into a language that uses free-form compounds. It is also possible to evaluate links that a word aligner has established between such compound pairs, based on the results from attempting such a segmentation process.

Three objectives were set for the algorithm: to verify part of the material, to clean out bad links and to test the general performance. The second objective, cleaning out bad links, would have been the most useful, as it would allow for an automated process to better an automatically generated bilingual dictionary. Unfortunately it failed. The first objective succeeded, but does not have the immediate advantages of the second objective. Still, it shows that a process as the one drawn up here can be useful, and one has to keep in mind that this is only an initial experiment.

Also introduced in this paper is the novel approach of silhouettes, which could feasibly be put to other uses than element extraction. One such thing is to pre-process a bilingual corpus before aligning it at the sub-sentence level. The sentences on the English side of the bilingual corpus could be compared word by word to all other sentences on the English side, to extract suitable link units. Being able to identify entire English free-form compounds could potentially help a word aligner in reducing the amount of partial links.

---

## Bibliography

- Ahrenberg, L., M. Andersson, and M. Merkel (1998). A simple hybrid aligner for generating lexical correspondences from parallel texts. In *Proceedings of COLING-ACL98*, Montreal, Canada, pp. 29–35.
- Dagan, I., K. Church, and W. Gale (1993). Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora*, pp. 1–8.
- Koehn, P. and K. Knight (2003). Empirical methods for compound splitting. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 187–194.
- Melamed, I. D. (1997). A word-to-word model of translational equivalence. In P. R. Cohen and W. Wahlster (Eds.), *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, Somerset, New Jersey, pp. 490–497. Association for Computational Linguistics.
- Och, F. J. and H. Ney (2000). Improved statistical alignment models. In *ACL00*, Hongkong, China, pp. 440–447.
- Sågvalld Hein, A. (1999). The PLUG-project: Parallel corpora in Linköping, Uppsala, Göteborg. aims and achievements. In *Working Papers in Computational Linguistics & Language Engineering 16*, Sweden. Department of Linguistics, Uppsala University.
- Sågvalld Hein, A., E. Forsbom, J. Tiedemann, P. Wijnitz, I. Almqvist, L.-J. Olsson, and S. Thaning (2002). Scaling up an mt prototype for industrial use. databases and data flow. In *Proceedings of LREC 2002. Third International Conference on Language Resources and Evaluation. Volume 5*, pp. 1759–1766.
- Saers, M. (2005). Example-based segmentation of swedish compounds in a swedish-english bilingual corpus. Master’s thesis, Department of Linguistics and Philology, Uppsala University, Sweden.
- Simard, M., G. Foster, and P. Isabelle (1992). Using cognates to align sentences in bilingual corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine translation (TMI92)*, Montreal, Canada, pp. 67–81.
- Tiedemann, J. (1999). Word alignment step by step. In *Proceedings of the 12th Nordic Conference on Computational Linguistics*, Norway.