

Automatic Recognition of Multi-Word
Expressions in Dutch
Bachelor's Thesis

Erik de Graaf, 3479080
Supervisor: prof. dr. Jan Odiijk

August 4, 2013

7,5 ECTS

1 Introduction

This Bachelor's thesis is built upon own research on the automatic recognition of multi-word expressions (MWEs) in Dutch. Multi-word expressions are phrases "whose exact and unambiguous meaning or connotation cannot be derived from the meaning or connotation of its components" (Choueka, 1988). An example is *de tent afbreken*, which literally translates to *breaking down the tent*, but is usually unrelated to tents when the phrase is used in its idiomatic sense. Being able to recognise MWEs is of great importance for natural language processing and artificial intelligence in general and a lot of research is done in on this topic (Al-Haj & Wintner, 2010; Bu, Zhu, & Li, 2010; Odijk, 2013). Because the meaning cannot be derived from the words, a database of MWEs will have to be constructed in order to build a computer capable of understanding this part of human language. Building a computer that possesses human-like intelligence is the ultimate goal for anyone studying artificial intelligence. Teaching a computer to successfully grasp the true meaning of MWEs would mean a good step forward for artificial intelligence technology. The University of Utrecht considers linguistics and computer science crucial parts of its programme on cognitive artificial intelligence. Together with psychology and philosophy these fields are the basis of the Cognitieve Kunstmatige Intelligentie programme at the university. This goal of this research and comparable research is compiling a list of MWEs and eventually their meanings. This study attempts to determine whether statistical distances can be used in the automatic recognition of MWEs. To my best knowledge this is the first study on Dutch MWEs that uses statistical distance methods.

At first, plans were to make a comparison between MWEToolkit (Ramisch, Villavicencio, & Boitet, 2010), an approach based on semantic domains and an approach based on statistical distance. I have previously worked with the semantic domains approach during the course *Woorden in Vertaling* (Words in Translation) (*Course Woorden in Vertaling*, 2012). During this course one had to identify MWEs by comparing how well words in the same semantic domain work together. This assignment was aimed at students with no programming experience and it therefore required manual lookups in the British National Corpus using the web-interface supplied by the Brigham Young University (*British National Corpus (BYU-BNC)*, n.d.). My choice was to use the ukWaC corpus (Baroni, Bernardini, Ferraresi, & Zanchetta, 2009) instead. Both the BNC and the ukWaC are tagged with part of speech tags, but a tree structure is not readily available for either corpus. The lack of tree structure prevented the students from filtering out words based on word relationship. This semantic domain based method is further explained in the recommendation part of this thesis on page 8. Other possible approaches are approaches based on fixed word order and approaches based on the rigidity of the plurality or singularity of words within an expression. Due to time limitations and computing power limitations it was decided to keep the focus of the this research on statistical distance (measured by PMI and salience) and the MWEToolkit.

2 Approach

To keep this research within reasonable size limits it was decided to only inspect MWEs that consist of a noun and a verb, where the noun is the head of the direct object of the verb. This will ensure that the noun actually is 'controlled' by the verb and not a noun that is not part of the direct object. The recognition is accomplished by comparing the results of an existing toolkit, the MWEtoolkit to a self-written Java program that calculates the Pointwise Mutual Information (PMI). The MWEtoolkit is used in order to be more easily able to identify the success of the PMI and salience methods. It consists of multiple Python scripts that are run over a corpus or frequency list. Salience is added to the comparison to avoid favouring low frequency phenomena such as typing errors and encoding errors. PMI and salience are useful measures because they rate word pairs on how often they occur together compared to the individual frequency of the words. It is expected that the pairs that make up an MWE appear as a pair relatively often, compared to how often these words appear on their own.

The corpus chosen for the research is the SoNaR500 part of the LASSY Large (Large Scale Syntactic Annotation of written Dutch)(van Noord et al, 2013) corpus. This corpus was chosen because, with 510 million words, it is the largest tree-tagged Dutch corpus. With help of the parse trees it is possible to only subtract nouns that are part of the direct object. The SoNaR part was chosen because it had a frequency list readily available and because 510 million words is plenty for this kind of research.

SoNaR comes with parsing trees in DACT's (Decaffeinated Alpino Corpus Tool)(de Kok, 2010) own format and in compact XML format. This research was done using DACT, which supports searches based on a simplified version of the XPath language, which itself is a subset of the XQuery language. This simplified version of XPath unfortunately limits the options the language offers so much that the compact XML files had to be used for the actual noun-verb pair extraction, which meant using non-indexed searches that take a lot of time to process. XQuery was used to take nouns that are in a relationship obj1 (direct object), directly under the node that also has a V (verb) with relationship hd (head). By using the following XQuery a list of 3.66 million pairs of verbs and nouns and their respective lemmas (non-inflected word form) was compiled.

```
$n in //node[@pos eq 'noun' and @rel eq 'obj1' and following-sib  
ling::*[1][self::node[@pos eq 'verb' and @rel eq 'hd']]]  
return  
( ($n|$n/following-sibling::*[1])/@lemma/string(), '\&\#xA;')
```

`/following-sibling` was replaced with `/previous-sibling` and `@lemma/string()` was replaced with `/@word/string()` to obtain the inflected word forms and to allow the reversed order occurrences. It is important to take notice of the occurrences of both V-NP and NP-V, because order has no meaning

in the LASSY trees, though words do occur in some order in the XML encoding.

Frequencies of the word pairs were calculated. This resulted in 699709 unique pairs of lemma entries with their frequency and 916779 unique pairs of inflected word entries with their frequency. The list of unique lemmas, consisting of pairs of *verb lemma* and *noun lemma*, was used as a candidate list.

Apart from a candidate list, a list with validated multi-word expressions was required. Such a list can help in determining whether candidates are MWEs or not. The flaw in this method is that these lists are not complete, which leads to MWEs being classified as non-MWE. Unfortunately this is the best method available next to manual validation, which takes an incredible amount of work and does not fit within the time limits set for a Bachelor's thesis. The validated MWE list was obtained by taking a list that was used in the process of creating DuELME(Grégoire, 2010) that was corrected by a linguist. DuELME is a lexicon of Dutch MWEs. Only noun-verb pairings were taken from the list. Unfortunately this list listed verbs in the singular form, whereas the lemmas in LASSY are in the plural form. A Java program was used to convert these verbs to their plural form, with help of the CGN-lexicon(Oostdijk, 2001). This lexicon lists a large amount of words, their pronunciation, their lemmas and more. The regular expression used for this was

```
. *? \ word \ WW (pv, tgw, ev) \ ( (? : [a-z] [a-z] + ) ) \
```

where *word* is the inflected word and a *()* block without preceding ** character indicates the saved lemma. Some manual editing took place to correct verbs that were not properly converted. The list of valid multi-word expressions contained 1010 entries. A list of verified incorrect entries was also created, which contained 758 entries. It is worth noting that this last list consists of candidates that were judged to be MWEs by different MWE identification tools but were manually confirmed not to be MWEs. For this reason this list is not a proper representation of the average non-MWE phrase and items in this list are more likely to get a high score than a random noun-verb combination would.

Both the MWEtoolkit and the self-written program are used to eliminate 698709 entries to remain with the top 1000. The correctness of those 1000 entries is then checked by seeing if they appear in the validated multi-word expressions list and the list with verified incorrect entries. The amount of times a noun-verb combination appears in both the top 1000 and the other lists is calculated. The calculated values are true-positive, false-positive and unknown. The unknown value is the amount of times a combination appears in neither the true nor the false list. It is not possible to obtain false-positives, because the list of validated multi-word expressions is only a small subset of the actual list of multi-word expressions, which doesn't exist yet. If the full list did exist this research would be pointless, as this research is an attempt at compiling such a list. The number of true-negatives is not calculated, because neither approach produces a list of

non-multi-word expressions. A list of non-multi-word expressions could be compiled by setting a threshold based on PMI, salience and MWEtoolkit score and labelling every phrase that does not meet the threshold as a non-MWE.

In an attempt at achieving better results the effectiveness of salience is checked in the same manner. Because of a lack of matches between the top 1000 lists and the validated MWE list a different way of checking effectiveness is used. This method checks for each validated MWE how high it appears in the full sorted list of candidate MWEs, based on PMI and salience. It is expected that the 1010 validated MWEs appear relatively high in the sorted candidate lists output by the MWEtoolkit, the PMI method and the salience method. If a method does not work as a method of identifying MWEs it is expected that the results resemble results of a method picking random places in the candidate list. This is recognisable by a score close to $\frac{699709}{2} \approx 349855$ after averaging.

3 Results

The two methods used were the MWEtoolkit sorted by Pointwise Mutual Information (PMI) and the self-written program. PMI is calculated with the following formula;

$$\text{pmi}(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)}$$

where $\text{pmi}(x; y)$ is the PMI score of x and y , $p(x, y)$ is the probability that x co-occurs with y , $p(x)$ and $p(y)$ are the probabilities that a certain word is x and y , respectively.

As the probabilities are unknown they had to be estimated. The probabilities of x and y have been estimated using two slightly different methods. For *Own PMI small* only the occurrences of x in the candidate list were counted, for *Own PMI large* the occurrences of x in the full 510 million word corpus were taken into account. The *small* method is not commonly used in comparable research, but it provides some interesting extra statistics.

By looking up the top 1000 candidates in the verified MWE list and the verified non-MWE list the following results were obtained. The *unknown* column lists the amount of pairs not found in either list.

Method	True-positive	False-positive	Unknown
MWEtoolkit PMI	3	5	992
Own PMI small	1	0	999
Own PMI large	1	4	995

No definite conclusion can be found using these data, as the number of entries that do not appear in either list accounts for 99% for each of the approaches. The poor results can be explained by looking through the 1000 suggested multi-word expressions of each approach.

The MWEtoolkit approach has a clear preference for the verb *hebben* (*to have*); 727 out of 1000 matches have *hebben* as verb. The verbs are in almost every case accompanied by a non-common noun like *klein-links* or *k3* or by a noun that contains spelling or encoding errors like *«jurist, studenten-hacker* and *b][color=pink]dat*.

The Own PMI approaches have less problems with *hebben*, with only 464 matches for the PMI large method and 3 matches for PMI small method. Own PMI small shows relatively many entries that include capital letters, and both Own PMI lists show large amounts of uncommon words or misspellings, much alike MWEtoolkit’s list. Neither list shows signs of many actual multi-word expressions, which suggests that the 99% unknown might very well mean close to 99% false positives. Verifying this is a time-consuming process, as it needs to be done manually.

To deal with the high amount of encoding errors and typing errors in the top 1000s the ranking method *salience* is used. This method multiplies the PMI by $p(x, y)$, giving rare occurrences a relatively low score and frequent occurrences a higher score. The resulting formula is

$$\text{salience}(x; y) \equiv \log \frac{p(x, y)^2}{p(x)p(y)}$$

This yields the following results:

Method	True-positive	False-positive	Unknown
Salient small	4	0	996
Salient large	7	6	987

These results are a slight improvement over the PMI results, but it is still not possible to tell whether the method is successful in determining MWEs or not.

In order to work around this issue, a different way of testing the rate of success was used. This method iterates both the confirmed-MWE list and the confirmed-non-MWE list and finds the position of the MWEs in the lists sorted by PMI/Salience/MWEtoolkit. The sum of these positions is taken. Ideally a positive hit would appear high in the list and therefore the sum of the positions for the positive list should be relatively low.

Method	Positive	Negative	Pos.avg.	Neg.avg.	Pos. - Neg.
MWEtoolkit PMI	300977688	221300648	385374	395886	-10512
Own PMI small	227317547	157770625	291059	282237	8822
Own PMI large	252619583	175879652	323457	314633	8824
Saliency small	192624130	143335038	246637	256413	-9776
Saliency large	150465118	110829503	192657	198263	-5606

Not all of the pairs in the confirmed and non-confirmed lists were found. The *average* columns are the average only over the pairs that were found. The last column is the average of the positives found subtracted by the average of the negatives. In this column negative numbers are the expected result. A negative result indicates that the verified MWEs appear higher than the verified non-MWEs. As every file contains exactly the same candidates it is not relevant that a few of the pairs did not have a match. It is worth noting that from a method that works at random values around $\frac{699709}{2} \approx 349855$ would be expected in the *average* columns. The conclusion from this is that the MWEtoolkit, even though it does have a large negative result in the last column, scores worse than chance. The PMI measure shows scores that are slightly better than random, but has a large positive number in the last column. This means it considered non-MWEs to be more likely than verified MWEs to be MWEs. The saliency method scores above chance and has acceptable results after subtraction of the average negative score.

It is difficult to arrive at definite conclusions from the *Pos. - Neg.* data. One could argue that the negative list has a lot of expressions that look like MWEs and because of that comparing *Pos. avg.* and *Neg. avg.* is not a useful measure. These values being rather close to zero does not necessarily mean these methods of finding MWEs are useless, they only indicate that the methods have a hard time distinguishing the verified MWEs from the verified non-MWEs. This is disappointing, but it does fall within expectations. Because the non-MWE list was compiled using a different MWE identification method and had to be manually verified not to be MWEs, it is not very surprising the methods used in this research are not successful distinguishing the verified MWEs from the 'look-alikes' either. The most important statistics are in the *Pos. avg.* column, as these indicate how high a positive match appears on average. Finding MWEs using PMI and the MWEtoolkit's version of PMI turns out not to work well. Saliency produces results that are better than chance, but that are still poor.

Conclusion and recommendations

None of the approaches was very successful in finding multi-word expressions in the LASSY corpus. The high percentage of hits that are not in the verified lists of multi-word expressions makes it very difficult to tell the difference in performance of the methods. A solution would be to manually check the found multi-word expressions and count them, but this would require a lot of time and is therefore not feasible for this Bachelor's thesis. A different way that was used to check the results is by looking up verified MWEs in lists sorted by likelihood of being an MWE. This does yield some interpretable results, but these results were quite disappointing.

The research in this thesis could have been improved in a couple of ways. The first way would have been to use a different statistical method. PMI values words that occur only a few times in the corpus too highly. Those words are usually encoding errors or spelling errors that are not necessarily parts of multi-word expressions. Saliency makes an attempt at solving this problem, and does so successfully, but it does not clean out all of the rare words. Saliency does show improved results, but the results it produces are still not very interesting. PMI and saliency might not be as suitable for determining whether a combination is an MWE as anticipated.

The method chosen for this thesis is only one of many. One of those methods was mentioned in the introduction of this thesis; comparing the semantic domains of the verbs and nouns. This would be done by grabbing a small number of words that are semantically close to the noun, a small number of words semantically close to the verb, and seeing how often those appear together in the corpus. There are multiple ways to find words that are semantically close to the words that are tested. The method used for the *Woorden in Vertaling* course (*Course Woorden in Vertaling*, 2012) assumes that words are likely to be in the same domain when they appear with a conjunction word such as *and* between them. A regular expression for obtaining the similar words would be `word\sand\s(?:[a-z][a-z]+)`, where *word* is the word that the semantic neighbours are searched for. `\s` is the regular expression way of denoting a space or tab character and the string between the parentheses takes care of grabbing all alphabetical characters that follow. This method is not suitable for tree structures although an adaptation to XQuery would be possible. A different method is looking up the words in WordNet (Miller, 1995) (for English words) or Cornetto (Vossen, Maks, Segers, & VanderVliet, 2008) (for Dutch words) and look for synonyms. The semantic domain method allows one to establish how likely it is that the verb and the noun appear in the same sentence. Low likelihood but many appearances of the actual combination would mean it is likely that the combination is a multi-word expression. A similar method is introducing a third semantic domain and seeing how well both words combine with the third one.

Other methods are the fixed word order method and the singularity/plurality method. The fixed word order method checks if certain words always or very often occur in the same word order. If they do, they are more likely to be (part of) an MWE. In noun-verb multi-word expressions the noun can sometimes appear in both singular and plural form, but not in all the cases. An example of a valid multi-word expression is *de plaat poetsen*, whereas changing the noun to its plural form would make the multi-word expression invalid (*de platen poetsen*). Each of these methods has its own flaws, it is likely that multiple methods have to be combined to achieve proper results.

What I have learned

Writing this thesis meant a good introduction to tree-tagged corpora for me. Working with such large sets of data is significantly more difficult than I had anticipated. This is partly because it required me to learn the workings of XML, XPath and XQuery, and partly because it introduces problems with memory limitations of programs and memory limitations of the machine that I worked with. I now have some more knowledge in these areas and I have gained important experience in programming efficiently. Most programming was done with short Perl scripts, a language which I was largely unfamiliar with before I started working on the research. I would not yet call myself a good Perl programmer, but I have definitely made progress there. The rest of the scripts/programs were written in Java, a language I already was familiar with. I have expanded my knowledge on multi-threaded programming, programming disk I/O efficiently and programming RAM efficiently. I also gained some experience with the GNU/Linux terminal, nano and vim. Using the Linux terminal was necessary because most text editors on Windows are very limited or do not work well with extremely large text files.

Altogether, working on this thesis was a very valuable experience that will be of good use in studies and research I will do in the field of computational linguistics in the future. My interest in linguistics, informatics, artificial intelligence and big data has grown and I am looking forward to learning a lot more in these fields.

References

- Al-Haj, H., & Wintner, S. (2010). Identifying multi-word expressions by leveraging morphological and syntactic idiosyncrasy. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)* (pp. 10–18). Beijing, China: Coling 2010 Organizing Committee.
- Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E. (2009). The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3), 209–226.
- British National Corpus (BYU-BNC)*. (n.d.). Retrieved from <http://corpus.byu.edu/bnc/>.
- Bu, F., Zhu, X., & Li, M. (2010). Measuring the non-compositionality of multiword expressions. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)* (pp. 116–124). Beijing, China: Coling 2010 Organizing Committee.
- Choueka, Y. (1988). Looking for needles in a haystack or locating interesting collocational expressions in large textual databases. In *Riao 88:(recherche d'information assistée par ordinateur)* (pp. 609–623).
- Course woorden in vertaling*. (2012). Retrieved from https://www.osiris.universiteitutrecht.nl/osistu_ospr/OnderwijsCatalogusSelect.do?taal=en&selectie=cursus&collegejaar=2012&cursus=201000050.
- Geyken, A. (2004). Bootstrapping a database of German multi-word expressions. In N. Calzolari et al. (Eds.), *Proceedings of the fourth international conference on language resources and evaluation (lrec'04)*. Lisbon, Portugal: European Language Resources Association (ELRA).
- Grégoire, N. (2010). DuELME: a Dutch electronic lexicon of multiword expressions. *Language Resources and Evaluation*, 44(1-2), 23–39.
- Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39–41.
- de Kok, D. (2010). *Dact [decaffeinated alpino corpus tool]*. Retrieved from <http://rug-compling.github.com/dact>.
- van Noord et al. (2013). Large scale syntactic annotation of written Dutch: Lassy. In *Essential speech and language technology for Dutch* (pp. 147–164). Springer.
- Odiijk, J. (2013). Identification and lexical representation of multiword expressions. In *Essential speech and language technology for Dutch* (pp. 201–217). Springer.
- Oostdijk, N. (2001). The design of the spoken Dutch corpus. *Language and Computers*, 36(1), 105–112.
- Ramisch, C., Villavicencio, A., & Boitet, C. (2010). Multiword expressions in the wild? The mwetoolkit comes in handy. In *Coling 2010: Demonstrations* (pp. 57–60). Beijing, China: Coling 2010 Organizing Committee.
- Spyns, P., & Odiijk, J. (2013). *Essential speech and language technology for Dutch: Results by the STEVIN-programme*. Springer.
- Vossen, P., Maks, I., Segers, R., & VanderVliet, H. (2008). Integrating lexical units, synsets and ontology in the Cornetto database. In N. Calzolari et

- al. (Eds.), *Proceedings of the sixth international conference on language resources and evaluation (lrec'08)*. Marrakech, Morocco: European Language Resources Association (ELRA).
- Weller, M., & Heid, U. (2010). Extraction of German multiword expressions from parsed corpora using context features. In N. Calzolari et al. (Eds.), *Proceedings of the seventh international conference on language resources and evaluation (lrec'10)*. Valletta, Malta: European Language Resources Association (ELRA).