MWE-Finder: a Demonstration

Jan Odijk, Martin Kroon, Tijmen Baarda, Ben Bonfil, Sheean Spoel
Utrecht University
Utrecht, the Netherlands
j.odijk@uu.nl, m.s.kroon@uu.nl, t.c.baarda@uu.nl, b.bonfil@uu.nl, s.j.j.spoel@uu.nl

Abstract

This paper introduces and demonstrates MWE-Finder, an application to search for flexible multiword expressions (MWEs) in Dutch text corpora, starting from an example. If the example is in canonical form, the application automatically generates three queries to search for sentences that contain an occurrence of the MWE and thus enables efficient analysis of its properties. The application offers canonical forms for more than 11k MWEs. Searching is done in treebanks, so the grammatical structure of the sentences is taken into account.

Keywords: multiword expressions, Dutch, MWE Finder, GrETEL 5, automatic query generation, searching for multiword expressions

1. Introduction

Many multiword expressions (MWEs) are flexible in the sense that their components can have different forms, can occur in different linear orders, or may not be contiguous, with other words appearing between elements of the MWE. This makes searching for such MWEs in large text corpora difficult. What is needed is a search system that can take all this flexibility into account.

In this paper we present such a system, called MWE-Finder. MWE-Finder enables a user to find occurrences of a multiword expression in a large Dutch text corpus. MWE-Finder is intended as a research tool for any linguist or lexicographer interested in research into multiword expressions, in particular *flexible* multiword expressions.

MWE-Finder (Odijk et al., 2023) can be used to address the task of MWE *identification* (in the sense of (Constant et al., 2017)): by using MWE-Finder a researcher can find occurrences of a given MWE easier and in a more reliable way than with other search applications. This will stimulate research into individual MWEs, their variants and properties, and their frequencies, thereby facilitating research into MWEs in general. The underlying software forms a good basis for software to automatically annotate large text corpora for MWEs, which not only may be beneficial for linguistic research but also for a variety of natural language processing tools dealing with MWEs.

MWE-Finder uses the DUtch CAnonicalised Multiword Expressions (DUCAME) resource to suggest MWEs to the user. DUCAME is a resource containing more than 11,000 MWEs for the Dutch language in canonical form.

The organisation of this paper is as follows. We begin with a brief introduction of the notion multiword expression (Section 2). MWE-Finder is presented in Section 3. We will end with conclusions and plans for future work (Section 4).

2. Multiword expressions

A MWE is a word combination with linguistic properties that cannot be predicted from the properties of the individual words or the way they have been combined by the rules of grammar (Odijk, 2013).[1]
A word combination can, for example, have an unpredictable meaning (*de boeken neerleggen*, lit. 'to put down the books', meaning 'to declare oneself bankrupt'), an unpredictabe form (e.g. *ter plaatse* 'on location', with idiosyncratic use of *ter* and *e*-suffix on the noun), or it can have only limited usage (e.g. *met vriendelijke groet* 'kind regards', used as the closing of a letter).

Words of a MWE need not always be fixed. This can be illustrated with the Dutch MWE *de boeken neerleggen* 'to declare oneself bankrupt'. The verb *neerleggen* in (1) can occur in all of its inflectional variants (e.g., past participle in (1a), infinitive in (1b), and past tense singular in (1c) and (1d)), and with the separable particle *neer* attached to it (1a, 1b) or separated (1c, 1d). MWEs do not necessarily consist of words that are adjacent, and the words making up a MWE need not always occur in the same order. This expression allows a canonical order with contiguous elements (as in (1a)), but it also allows other words to intervene between its components (as in (1b)), as well as permutations of its component words (as in (1c)), and combinations of permutations and intervention by other words that are not components of the MWE (as in (1d)):

(1) a. Saab heeft gisteren **de boeken**
Saab has yesterday the books
neergelegd.
down.laid

[1] For a similar but slightly different definition see (Sag et al., 2001).

'Saab declared itself bankrupt yesterday.'

b. Ik dacht dat Saab gisteren **de** **boeken** wilde **neerleggen**.
I thought that Saab yesterday the books wanted down.lay
'I thought Saab wanted to declare itself bankrupt yesterday.'

c. Saab **legde** **de** **boeken** **neer**.
Saab laid the books down
'Saab declared itself bankrupt.'

d. Saab **legde** gisteren **de** **boeken** **neer**.
Saab laid yesterday the books down
'Saab declared itself bankrupt yesterday.'

In addition, certain MWEs allow for (and require) controlled variation in lexical item choice, e.g. in expressions containing bound anaphora such as *zijn geduld verliezen* 'to lose one's temper', where the possessive pronoun varies depending on the subject (cf. *Ik verloor mijn/*jouw geduld; jij verloor *mijn/jouw geduld*, etc.), exactly as the English expression *to lose one's temper*. Of course, not every MWE allows all of these options, and not all permutations of the components of a MWE are well-formed (e.g. one cannot have **Saab heeft neergelegd boeken de.* lit. 'Saab has downlaid books the.').

This flexible nature of such MWEs makes it difficult to reliably search for such expressions in text corpora. Standard search engines such as Google do not enable the user to systematically search for different word forms of the same lemma. Search applications for Dutch such as OpenSoNaR (van de Camp et al., 2017; de Does et al., 2017) or Nederlab (Brugman et al., 2016) can do this, but it is difficult to formulate a query allowing different orders and interspersed irrelevant words, and the results of such a query will be unreliable. At best, one will find all instances but at the same time also many cases where all the component words occur but do not make up a MWE. One should be able to search for flexible MWEs in such a way that their grammatical structure is taken into account. This can be done in a treebank, and MWE-Finder enables searching for MWEs in a treebank.

3. MWE-Finder

MWE-Finder enables a user to search for occurrences of a MWE in a treebank based on an example MWE if this example MWE is in canonical form. MWE-Finder provides a list of more than 11k MWEs in canonical form. This canonical form will not be discussed here.[2]

MWE-Finder is embedded in GrETEL, an existing web application for searching Dutch treebanks (Augustinus et al., 2012, 2017; Odijk et al., 2018). The distinguishing feature of GrETEL is its query-by-example feature. In its regular search mode, it leads the user through a number of steps to get from an example sentence to search results and analysis of the search results:

1. **Example:** A user can enter a natural language example that illustrates the construction the user is interested in.

2. **Parse:** The Alpino parser (Bouma et al., 2001; van der Beek et al., 2002) parses the example sentence. The parse is represented in XML.

3. **Matrix:** The user indicates which words of this example are crucial for the construction, and how each word should be generalised from. Based on this the parse tree of the example sentence is transformed into a query (GrETEL uses XPath).

4. **Treebanks:** The user can select one or more treebanks to search in.

5. **Results:** The XPath query is applied to the selected treebank(s) and the results are provided as a list of sentences with matches.

6. **Analysis:** The results can be further analysed in a graphical interface to a pivot table for properties of the nodes in the query in combination with any available metadata.

A second important feature of GrETEL is that one can upload one's own text corpus, which is then automatically parsed and made available as a treebank to search in.

MWE-Finder is part of version 5 of the web application GrETEL, available in a first version since the end of 2022.[3] Thanks to this integration, MWE-Finder has access to all GrETEL features, and supports all treebanks that are included in GrETEL as well as its possibility of uploading one's own text corpora.

3.1. Illustration

MWE-Finder partially mimics the structure of GrETEL's main search functionality. It distinguishes the following steps: *Canonical Form* (cf. GrETEL's *Example* step), *Treebanks*, *Results*, and *Analysis*. It currently lacks the *Parse* step and the *Matrix* step.

[2] The data and documentation can be found on https://surfdrive.surf.nl/files/index.php/s/2Maw8O0QTPH0oBP. See (Jan Odijk, 2023).

[3] https://gretel5.hum.uu.nl.

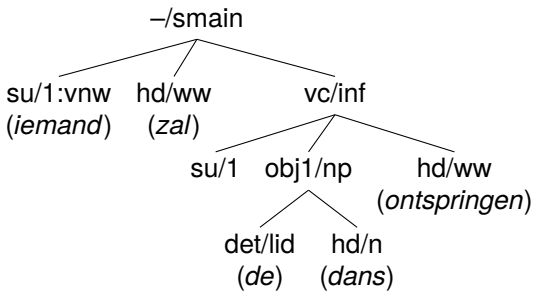

Figure 1: Syntactic structure of *iemand zal de dans ontspringen* (simplified).

MWE-Finder enables the user to enter a MWE example, just like GrETEL, though it must be in canonical form. The user thereby implicitly formulates a hypothesis about the properties of this MWE. Queries are automatically generated on the basis of this canonical form. The presence or absence of annotations on the example determine how the system generalises from this example, so these annotations can be seen as a different way of implementing the *Matrix* step.

The MWEs contained within DUCAME have been included in a drop-down list and are directly searchable within the MWE-Finder. The user can also enter a new MWE, provided that it complies with the conventions for MWE canonical forms.

As a concrete example, suppose that the user enters the canonical form (2):

(2) iemand zal de **dans** **ontspringen**
someone will the dance escape
'Someone will get off scot-free.'

This canonical form is parsed by the parser in MWE-Finder, resulting in the syntactic structure in Figure 1.

After the MWE has been parsed, the system automatically generates three queries to search for occurrences of this MWE in a treebank that correspond to different levels of agreement between the MWE and the sentences of the corpora. These are the *MWE query*, the *near-miss query*, and the *major lemma query*.[4]

The MWE query searches for occurrences of the MWE. The near-miss query searches for the MWE but allows unexpected determiners and modifiers (and thus returns a superset of the MWE Query). The major lemma query searches for utterances that contain the major lemmas of the MWE (and returns a superset of the near-miss query). The procedure to derive these queries has been described in (Odijk et al., to appear 2024).

Next, the user can select the treebank or treebanks that the query should be applied to. Once selected, the application switches to the *Results* view where query results are displayed as they arrive from the server. In that view, the user can also switch between the different queries for the chosen MWE or choose to exclude results of finer-grained queries. It is also possible to inspect or manually change the automatically generated XPath queries and retrieve new results.

In the *Results* view, users can also look at the parse trees for results or toggle extra context (one preceding sentence, one following sentence) to better analyze the occurrences found, just like in GrETEL. The automatically generated MWE query is shown in Figure 2.

When we apply this query to the Mediargus treebank,[5] MWE-Finder finds 1158 hits in over 103 million sentences.

The near-miss query is given in Figure 3. It finds 1271 hits in the Mediargus treebank.

If we exclude the results of the MWE query from the results of the near-miss query, which is an option offered by MWE-Finder, we quickly see in the 131 remaining hits that *de dans ontspringen* occurs in variants not predicted by the hypothesis implicitly formulated in the canonical form that we started with. We list some examples of phrases that the word *dans* occurs with:

other determiners None, *die* 'that', *zijn* 'his';

adjectival modifiers *gerechtelijke* 'judicial', *fiscale* 'fiscal', *politieke* 'political';

PP modifiers *van de bedreigden* 'of the threatened ones', *van de sociale verkiezingen* 'of the social elections'.

All of this clearly suggests that the canonical form that we started with was too strict. We must allow for modification of the MWE component *dans* and the article *de* is not a component of the MWE. A

[4] Note that MWE-Finder can identify potential occurrences of a MWE in a treebank. It cannot determine for an expression that is ambiguous between a literal and an idiomatic reading which of these alternative readings is applicable in a specific sentence.

[5] A large treebank with Flemish newspaper text created by Kris Heylen from KU Leuven in 2009.

```
//node[
      node[@rel="obj1" and @cat="np" and count(node)=2 and
          node[@rel="det" and @cat="detp" and count(node)=1 and
              node[@lemma="de" and @rel="hd" and @pt="lid" and
                   @lwtype="bep"]
              ] and
          node[@lemma="dans" and @rel="hd" and @pt="n" and
               @ntype="soort" and (@genus="zijd" or @getal="mv") and
               @getal="ev" and @graad="basis"]
          ] and
      node[@lemma="ontspringen" and @rel="hd" and @pt="ww"]
      ]
```

Figure 2: The MWE query for *de dans ontspringen*.

```
//node[
      node[@rel="obj1" and @cat="np" and
          node[@lemma="dans" and @rel="hd" and @pt="n" and
               @ntype="soort" and (@genus="zijd" or @getal="mv")]
          ] and
      node[@lemma="ontspringen" and @rel="hd" and @pt="ww"]
      ]
```

Figure 3: The near-miss query for *de dans ontspringen*.

better canonical form for this MWE would be *iemand zal 0de *dans ontspringen*, which explicitly allows modification of *dans*, and explicitly states that the determiner *de* is not a component of the MWE. Indeed, the MWE query derived from this canonical form finds 1271 hits, the same number as the near-miss query for the original canonical form. In this way, we can improve upon an initial canonical form mainly based on native speaker intuitions by systematically taking into account corpus data. MWE-Finder makes this possible in a very efficient and user-friendly way.

The major lemma query (see Figure 4) finds 1309 hits. If we exclude the results of the near-miss query, we have to inspect 38 examples. These are mostly valid instances of the MWE *de dans ontspringen* that have been wrongly parsed by Alpino, but we also find a variant of the MWE, viz. (3) for which we can now add a canonical form to DUCAME.

(3) iemand zal **aan** de **dans** **ontspringen**
someone will to the dance escape
'Someone will get off scot-free.'

In this way, a linguist or lexicographer can easily and efficiently investigate the properties of Dutch MWEs, and improve the description of Dutch MWEs.

Finally, there is the analysis step, which is currently identical to the one in GrETEL but we are working on a dedicated analysis component for MWEs, so that the search results can be investigated even more efficiently.

4. Concluding Remarks

We have introduced and demonstrated MWE-Finder, an application that, we submit, is very useful for linguistic and lexicographic research into MWEs, We hope to make this tool even more useful by extending it with an analysis component dedicated to MWEs.

Acknowledgements

WE thank the anonymous reviewers for comments on an earlier version of this paper, which has led to many improvements. Of course, all errors are ours.

5. Bibliographical References

Liesbeth Augustinus, Vincent Vandeghinste, and Frank Van Eynde. 2012. Example-based treebank querying. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3161–3167, Istanbul, Turkey. European Language Resources Association (ELRA).

Liesbeth Augustinus, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde. 2017. GrETEL: A tool for example-based treebank mining. In Jan Odijk and Arjan van Hessen, editors,

```
//node[@lemma="dans" and @pt="n"]
/ancestor::alpino_ds/node[@cat="top" and
 descendant::node[@lemma="ontspringen" and @pt="ww"]]
```

Figure 4: The major lemma query for *de dans ontspringen*.

CLARIN in the Low Countries, chapter 22, pages 269–280. Ubiquity, London, UK. License: CC-BY 4.0.

Gosse Bouma, Gertjan van Noord, and Robert Malouf. 2001. Alpino: Wide-coverage computational analysis of Dutch. *Language and Computers*, 37(1):45–59.

Hennie Brugman, Martin Reynaert, Nicoline van der Sijs, René van Stipriaan, Erik Tjong Kim Sang, and Antal van den Bosch. 2016. Nederlab: Towards a single portal and research environment for diachronic Dutch text corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1277 – 1281, Paris, France. European Language Resources Association (ELRA).

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

J. de Does, J. Niestadt, and K. Depuydt. 2017. Creating research environments with BlackLab. In Jan Odijk and Arjan van Hessen, editors, *CLARIN in the Low Countries*, chapter 19, pages 245–257. Ubiquity, London, UK. License: CC-BY 4.0.

Jan Odijk. 2013. Identification and lexical representation of multiword expressions. In P. Spyns and J.E.J.M Odijk, editors, *Essential Speech and Language Technology for Dutch. Results by the STEVIN-programme*, Theory and Applications of Natural Language Processing, pages 201–217. Springer, Berlin/Heidelberg.

Jan Odijk, Martin Kroon, Tijmen Baarda, Ben Bonfil, and Sheean Spoel. to appear 2024. MWE-finder: Querying for multiword expressions in large Dutch text corpora. In Voula Giouli and Verginica Barbu Mititelu, editors, *Multiword expressions in lexical resources. Linguistic, Lexicographic and Computational perspectives*, Phraseology and Multiword Expressions. Language Science Press, Berlin.

Jan Odijk, Martijn van der Klis, and Sheean Spoel. 2018. Extensions to the GrETEL treebank query application. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16)*, pages 46–55, Prague, Czech Republic.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2001. Multiword expressions: A pain in the neck for NLP. *LinGO Working Paper*, 2001-03.

Matje van de Camp, Martin Reynaert, and Nelleke Oostdijk. 2017. WhiteLab 2.0: A web interface for corpus exploitation. In Jan Odijk and Arjan van Hessen, editors, *CLARIN in the Low Countries*, chapter 19, pages 231–243. Ubiquity, London, UK. License: CC-BY 4.0.

Leonoor van der Beek, Gosse Bouma, and Gertjan van Noord. 2002. Een brede computationele grammatica voor het Nederlands. *Nederlandse Taalkunde*, 7:353–374.

6. Language Resource References

Jan Odijk. 2023. *DUCAME: DUtch CAnonicalised Multiword Expression Database*. Utrecht University. Utrecht University, 3.0. [link].

Jan Odijk and Martin Kroon and Tijmen Baarda and Ben Bonfil and Sheean Spoel. 2023. *MWE-Finder. Application to identify Dutch MWEs*. Utrecht University. Utrecht University. [link].