The ANW: an online Dutch Dictionary

Carole Tiberius & Jan Niestadt
Instituut voor Nederlandse Lexicologie (INL)
Leiden
The Netherlands
{carole.tiberius,jan.niestadt}@inl.nl

Abstract
The *Algemeen Nederlands Woordenboek* (ANW) is an online scholarly dictionary of contemporary standard Dutch. It offers a wide range of search possibilities supporting both semasiological and onomasiological queries. This paper discusses the search application of the ANW dictionary.

1. The ANW dictionary

The *Algemeen Nederlands Woordenboek* is a comprehensive online scholarly dictionary of contemporary standard Dutch in the Netherlands and in Flanders, the Dutch speaking part of Belgium (Moerdijk 2004, 2008; Moerdijk, Tiberius & Niestadt 2008). The dictionary focuses on written Dutch and covers the period from 1970 onwards. The dictionary was conceived as an online dictionary right from the outset and offers a range of search possibilities supporting both semasiological and onomasiological queries. A demo version of the dictionary[1] was launched at the end of 2009. Consultation of the dictionary is free. Users are only required to register on their first visit.

2. Searching the ANW dictionary

Electronic dictionaries promise dynamic, proactive search via multiple criteria and via diverse access routes, but, often, they do not realise the full potential. Within the ANW, a range of search strategies is offered, from text understanding to text production. However, instead of using the traditional dichotomy between basic and advanced searches, the ANW opts for a more intuitive presentation where the search strategies are coupled to queries users may ask. That is, they are based on the starting point and goal of user queries. Four search options are distinguished:

a) Word → Meaning, i.e. search for information about a word or phrase;
b) Meaning → Word, i.e. search for a word starting from the meaning;
c) Features → Words, i.e. search for words with one or more shared features.
d) Examples, i.e. search for example sentences.

We believe that in this way, users have a better overview of what they can actually search for and will be more enticed to explore the various options. The different search options can be accessed through four clearly marked tabs at the left top corner of the interface (see Figure 1). We will briefly discuss these four search options below.[2]

[1] http://anw.inl.nl
[2] We intend to make an English user interface available. Although the dictionary is primarily targeted at Dutch users, we believe it may also be useful, for instance, for language learners who would benefit from an English interface.

Figure 1 Illustration of the different search options in the user interface

2.1 Word → Meaning

The first tab "word → meaning" (*woord → betekenis*) is the traditional search which allows the user to search for information about a word or phrase in the dictionary. It is offered on the start page of the dictionary (see also Figure 1). The search box is clearly marked and examples illustrate the possibility of using wildcards.

2.2 Meaning → Word

The second tab is for the onomasiological search. This allows users to look for a word that they have forgotten or it can be used to find out whether there is a word for a certain concept or not. For instance, what is the plastic or metal tag at the end of a lace called?[3] In order to assist the user, two alternative strategies are offered to arrive at an answer. First, users can search by giving a definition or a description or by summing up terms that spring to mind. Second, they can use a guided search, which is based on the semagrams[4] in the dictionary. In this case, they are asked to choose the category (is it a thing, a person, an animal, a vehicle, etc.?) of the word they are looking for. Once a category has been selected, a number of questions pop up on the screen which are related to the most prominent features of that semantic class. This is illustrated for the category 'person' in Figure 2. Based on answers to questions such as "What does this person do?", "What does this person look like?", "Where does this person live?", the computer arrives at the word the user is looking for.[5]

[3] In English this is called an 'aglet' (*nestel* or *malie* in Dutch).
[4] The ANW provides a twofold meaning description. In addition to definitions, there are semagrams, A semagram is a systematic representation of the knowledge associated with a word in a frame of slots and fillers. Semagrams have been described in more detail by Moerdijk (2008).
[5] Obviously, the functionality of this search option is limited in the demo version which contains 914 lemmas. It will become more interesting when more data is added.

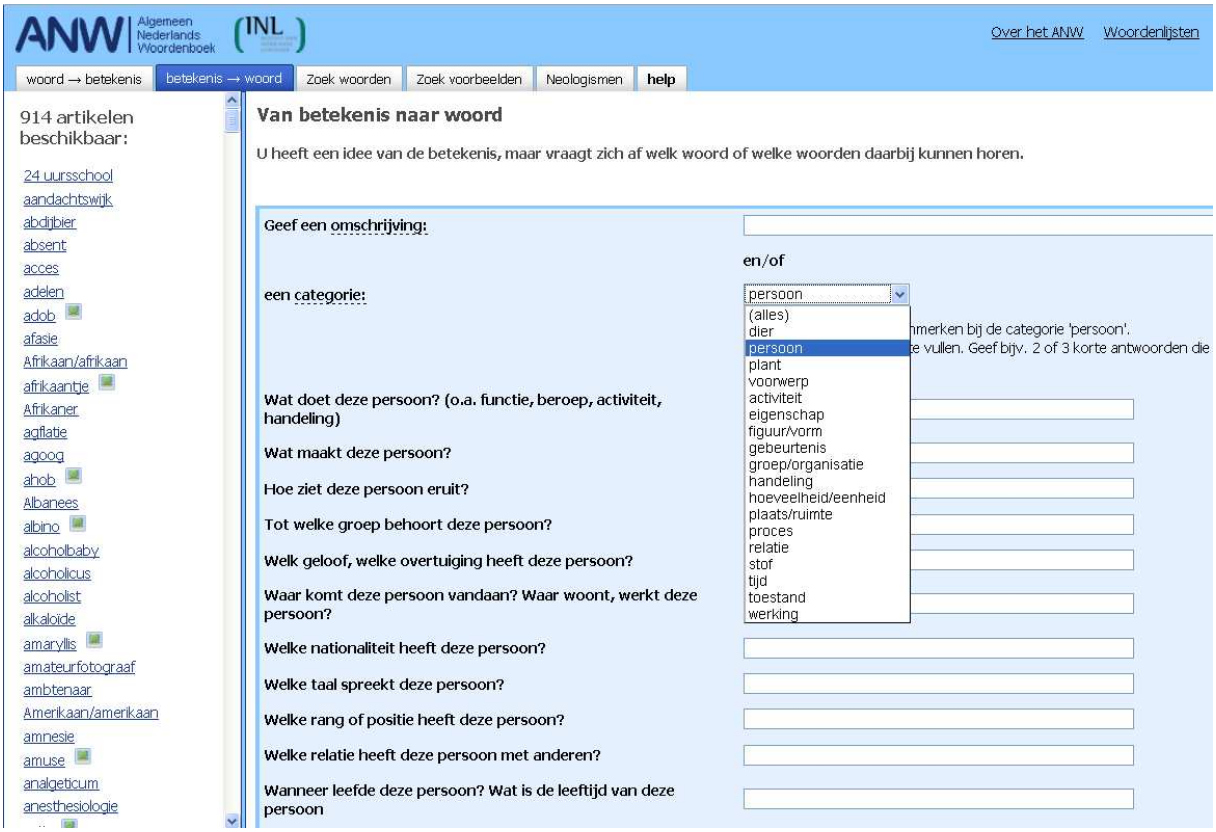


Figure 2 Illustration of search option: Meaning → Word

2.3 Features → Words

This search option is particularly relevant for language professionals. It enables them to gather words that share one or more identical features within the main dimensions of the ANW, e.g. orthography, morphology, meaning, combinatorics. In theory the user can search for all the elements and sub-elements that are available in the dictionary. This means that a total of nearly 200 features can be searched for. Although this is a very complex search option, undoubtedly offering the user a surplus of possibilities, we feel that if we manage to entice the user enough to experiment with it, he can easily become addicted. Puzzlers, for instance, can use this option to find words that begin and end with a particular letter, adding additional constraints such as domain or semantic class, helping them to solve their puzzles.

To assist the user in finding his way through the forest of criteria, they are presented in a structured way using a tree structure such as that of Windows Explorer with the advantage that users know immediately how to deal with them. In Figure 3 we illustrate this with an example of a simple query where we are looking for all words that begin with an *s* and consist of five syllables.[6] The user starts from an empty query screen and is asked to select criteria from the tree structure on the left. By default, the user searches for words, but it is also possible to search for proverbs or idioms. This will result in a tree structure with different criteria as only a subset of the criteria that can occur in a query for words apply to idioms and proverbs.

[6] This query results in words such as *sonjabakkeren* (‘go/be on a ‘Sonja Bakker diet’) and *sportverslaggever* (‘sports journalist’).

Figure 3 Illustration of search option Features → Words

All search options support the use of wild cards, and the input undergoes some linguistic analysis including stemming and removal of stop words. The results are ranked by relevance, but other sorting options are offered as well.

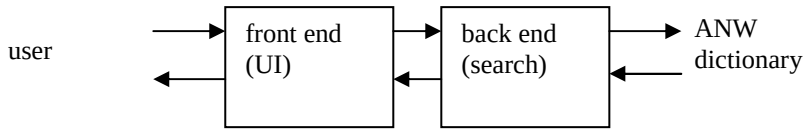
2.4 Example sentences

This option allows the user to search for example sentences based on a set of four criteria, i.e. word(s), author, source and date. For instance, a user could search for all example sentences with the words *koe* 'cow' and *schaap* 'sheep' in the period from 2000 – 2002 (date). The results can be sorted alphabetically (lemma) or chronologically (date).

3 The ANW application

The ANW application was built on the basis of a detailed Functional Design report (Moerdijk et al. 2008) in which the required search strategies and criteria are set out. This report was compiled by a team consisting of a lexicographer, a computational linguist and a software engineer. It took approximately nine person months to build the application.

The application is a servlet written in Java and runs in Apache Tomcat. It consists of two parts: a front end and a back end.



Front end

The front end is the part which runs in the browser of the user. It presents the user interface and allows the user to enter queries and to navigate through the result. The front end of the ANW application does not use Flash or similar technologies, using instead HTML and Javascript. The advantage is that the application "feels at home" in the browser and all browser functionalities (back button, bookmarking) continue to work as usual, following the user's expectations. There is full browser support at least for Internet Explorer 6/7/8, Firefox 3 and Safari 3.

Back end

The back end assists the front end by executing user queries, retrieving material, etc. It contains the search engine of the application. As part of the search engine, a separate query language was developed for the ANW, called FunQy (Niestadt et al. 2009). FunQy stands for "functional query language", which evaluates expressions to yield Lucene query objects.

Probably the easiest way to get a feel for this query language is to look at a few examples. A very basic query is to find words that have the word 'animal' in their definition text. The FunQy query looks like this:

```
Definition ~ 'animal'
```

FunQy supports wildcards, regular expressions and phrase search. It has special operators for stemming, fuzzy search and distance search:

```
Definition ~$ 'animal'
Definition ~~ 'animal'
Definition ~ 'animal' #3 'beast'
```

For convenience, fields may be combined using logical operators if one wishes to search more than one field (for example, definition and hyperonym). Therefore, the following two are equivalent:

```
Definition ~ 'animal' | Hyperonym ~ 'animal'

(Definitie | Hyperonym) ~ 'animal'
```

It is also possible to define symbols for later use. The following two queries in sequence are again equal to the above:

```
MyFavouriteFields = Definitie | Hyperonym;
MyFavouriteFields ~ 'animal'
```

Furthermore, functions can be defined. For example, it is possible to define the 'exclusive OR' in FunQy:

```
xor = function a, b -> (a | b) & !(a & b)
```

It is also easy to bind any custom Java code to FunQy. For example, we have a 'terms' function that chops user input into terms (words):

```
listOfWords = terms(userInput)
```

Some advanced features include passing functions as parameters and partial function instantiation. A simple 'standard library' written in FunQy provides operations such as filtering a list, mapping a list to a different list and combining items in a list.

Altogether, this makes FunQy a fairly complete functional programming language and a powerful tool for specifying search logic. An additional advantage is that this query language allows lexicographers, who know the dictionary data best, to experiment more and to help with

the fine-tuning of the search engine of the application. They can experiment with different parameters, including stemming and fuzzy matching. The 'search intelligence' for the specific searches we offer in the user interface resides in an application-specific script written in this language.

The ANW data is stored in a MySQL database. Apache Lucene is used for full text indexing. We combine the search results from Lucene with a hierarchical model of the dictionary article content in the MySQL database, so we can easily answer questions like "show me examples from senses with the word 'animal' in the definition" or "show me senses that contain examples that include the word 'animal' ". This hierarchical model is also used to show the full article with all search hits highlighted.

4 Conclusion

In this paper we have discussed the search application which was built for an electronic dictionary of Dutch, the ANW. It offers a range of search possibilities supporting both semasiological and onomasiological queries in a rather intuitive way. In this paper we have focused on the access strategies that are offered and on FunQy, the query language that was specifically developed for the project to facilitate implementation and future extensions to the search options offered by the ANW. Currently the demo version of the dictionary has just over 2000 registered users.

Acknowledgements

We would like to thank Fons Moerdijk, the previous chief-editor of the project, who played a major role in the design phase of the application. Thanks also to Matthew Baerman for proofreading the English.

References

Niestadt, Jan, Carole Tiberius & Fons Moerdijk (2009). Searching the ANW dictionary. Poster presented at eLexicography in the 21st century. Louvain-la-Neuve.

Moerdijk, Fons (2004). Het Algemeen Nederlands Woordenboek (ANW). *Nederlandse Taalkunde* 9, 175-182.

Moerdijk, Fons (2008). Frames and semagrams; Meaning description in the General Dutch Dictionary. in: *Proceedings of the Thirteenth Euralex International Congress, EURALEX 2008*. Barcelona.

Moerdijk, Fons, Carole Tiberius & Jan Niestadt (2008). Accessing the ANW Dictionary, in: *Proceedings of the Workshop on Cognitive Aspects of the Lexicon (COGALEX 2008)*, 18-24. Coling 2008 Organizing Committee. www.aclweb.org/anthology/W08-1903

Moerdijk, Fons, Jan Niestadt, Carole Tiberius & Michel Boekestein (2008). *Functioneel Ontwerp ANW op Internet*. Instituut voor Nederlandse Lexicologie, internal report.