Elaborating the parameterized Equivalence Class Method for Dutch

Nicole Grégoire

Uil-OTS, University of Utrecht
Trans 10, 3512 JK Utrecht, The Netherlands
Nicole.Gregoire@let.uu.nl

Abstract

This paper discusses the parameterized Equivalence Class Method for Dutch, an approach developed to incorporate standard lexical representations for Dutch idioms into representations required by any specific NLP system with as minimal manual work as possible. The purpose of the paper is to give an overview of parameters applicable to Dutch, which are determined by examining a large set of data and two Dutch NLP systems. The effects of the introduced parameters are evaluated and the results presented.

1. Introduction

MultiWord Expressions (MWEs) are used frequently in every day language, usually to express precisely ideas and concepts that cannot be compressed into a single word. MWEs can be defined as sequences of words that has linguistic properties not predictable from the individual components or the normal way they are combined (Odijk, 2004b).

MWEs form a serieus problem for many areas of language technology. Their unpredictable meaning and restrictions on syntactic variability makes them unsuitable for literal treatment. For successful handling of MWEs, both the grammar and the lexicon of an NLP system must be extended.

Our research focuses on making available a large number of lexical entries for MWEs for the use in various NLP systems. We aim at a databank of 5,000 expressions that meets the criterion of being highly theory- and implementation-independent. The method being developed is based on the so-called Equivalence Class Method (ECM) originally proposed by Odijk (2003).

The purpose of the current paper is to elaborate the parameterized ECM for Dutch, in particular to define a set of parameters suitable for Dutch and to determine to what extent they contribute to optimizing the parameterized ECM. The focus is on one specific type of MWE, viz. idioms, and Van Dale dictionary data are used as a basis for creating the testing material.

In the remainder of this section, I present my characterization of idioms and briefly describe the problem. Section 2 describes the ECM. Section 3 elaborates on the Parameterized ECM. Measurements are carried out in Section 4. I summarize the major conclusions in Section 5, and conclude this paper with some future work.

1.1. Idioms

In the literature, an idiom is not only defined in various ways, there is also a lot of variety in terms describing the phenomenon of idiomaticy (Everaert et al., 1995). Despite the many definitions, it is generally agreed that the meaning of an idiom cannot be derived from the meaning of its parts (Nunberg et al., 1994).

In this paper idioms are defined as MWEs headed by a verb (non-finite in the canonical form) with a fixed (or very limited) item selection and which meaning cannot be obtained compositionally from the meaning of its parts when used in isolation. Examples of some Dutch idioms are given in (1).

(1) a. het licht zien
the light see
'see the light'

b. over lijken gaan
across dead-bodies go
lit. 'go across dead bodies '
id. 'show no mercy'

c. naast zijn schoenen lopen
next-to his shoes walk
lit. 'walk next to his shoes'
id. 'be full of conceit'

1.2. Problem description

In order for idioms to participate in the syntax as normal expressions, i.e. allow for permutations, intrusions by other words and phrases, etc., it must be specified where and how the parts of the idiom must be realized and how they combine with free arguments. The way to account for this is to assign to an idiom the syntactic structure that it would have as a literal expression.[1] The problem, however, with syntactic structures in NLP systems is that they are highly system specific. This has been shown by Odijk (2004a) using the Rosetta machine translation system (Rosetta, 1994) as illustration. The Rosetta system requires, for idiomatic expressions, (1) reference to a highly specific syntactic structure, and (2) a sequence of references to lexical entries of the lexicon of the system. In this sequence the presence/absence of these references, the order in the sequence, and the references themselves are all particular to the Rosetta system. Highly specific representations are undesirable, since it requires effort in making such representations for each new NLP system again and reuse of significant effort is not possible.

No de facto standard for the lexical representation of MWEs currently exists. Various attempts have been made to develop a standard encoding for certain MWEs, espe-

[1] Idioms often have restrictions on their syntactic behavior additional to the ones on normal constructions. The syntactic flexibility of idioms is briefly addressed in the section Future work, but ignored in the rest of this paper.

Pattern	Comments
IDp1	Expressions headed by a verb taking a direct object NP that consists of a determiner and a singular noun.
IDp2	...

Table 1: An example of an idiom pattern description.

Pattern name	ICL	Example
IDp1	de plaat poetsen (lit. 'to polish the plate', id. 'to clear off')	Hij heeft de plaat gepoetst
IDp1	de boot missen (lit. 'to miss the boat', id. 'to miss the boat')	Hij heeft de boot gemist
IDp1	de kar trekken (lit. 'to pull the cart', id. 'to carry the load')	Hij heeft de kar getrokken

Table 2: List of idiom descriptions.

cially within the ISLE[2] and XMELLT[3] projects. Odijk argues that these attempts are unlikely to be successful, because the structures assigned to the MWEs are highly theory-dependent and even within one grammatical framework, there will be many differences from implementation to implementation. Since most syntactic structures are fully specified tree structures, they are difficult to create and maintain. Copestake et al. (2002) outlined an approach to represent MWEs in a form which can support precise HPSG, and which is also claimed to be reasonably transparant and reusable. Though their approach may work for certain types of MWEs, they fail to come up with a satisfying solution for representing idioms.
The parameterized ECM should offer a theory- and implementation-independent solution to the problem of lexical representation of idioms.

2. The Equivalence Class Method for idioms

Instead of describing the structure of an idiom, the ECM requires that it is specified which idioms have the same structure.
Odijk proposes that an idiom description should consist of the following parts:

1. An idiom pattern name: an identifier that uniquely identifies the structure of the idiom.
2. A list of idiom components (Idiom Component List: ICL).
3. An example sentence that contains the idiom.

The equivalence classes are defined with the help of the idiom patterns, i.e. idioms with the same pattern belong to the same equivalence class. The ICL takes the form of a sequence of strings, each string representing the lexicon citation of an idiom component. The order of the sequence is free, but the standard requires that the same order is used for each idiom in the same equivalence class. As for the example sentence, the standard requires that its structure should be identical for each example sentence within the same equivalence class.
Besides the idiom description, there must be a list of idiom pattern descriptions. Each idiom pattern description consists of two parts:

1. An idiom pattern.
2. Comments, i.e. free text in which the uniqueness of the pattern is described.

In Table 1 and 2 an illustration is given of the proposed standard. Table 1 shows one idiom pattern description and Table 2 shows three instances of the same equivalence class, i.e. with the same idiom pattern.
Given a class of idiom descriptions, representations for a specific theory and implementation can be derived. The procedure is that one instance of an equivalence class must be converted manually. By defining and formalizing the conversion procedure, the other instances of the same equivalence class can be converted in a fully automatic manner. In other words, having the equivalence classes consisting of idioms with the same pattern, it requires some manual work to convert one instance of each equivalence class into a system specific representation, but all other members of the same equivalence class can be done in a fully automatic manner.
A potential problem of the ECM as proposed is the risk that the number of equivalence classes will run into thousands of which the majority contains only a small number of idioms.[4] Since the ECM concentrates on minimizing the manual work when incorporating a large number of idioms in a specific system, the method will be less successful if there are many equivalence classes with only a few instances.
In order to reduce the number of equivalence classes and to increase the number of members within each equivalence class, Odijk (2004a) introduced the parameterized equivalence classes.

[2] www.ilc.cnr.it/EAGLES96/isle/
[3] www.cs.vassar.edu/∼ide/XMELLT.html
[4] This problem was also raised by Copestake et al. (2002), though not in relation to the ECM.

Category	PC	PC description	PV	PV description
determiner	*dbin*	binding type	DSB	subject bound
			DOB	object bound
noun	*ngen*	the gender of the noun	DE	definite article for masculine and feminine nouns
			HET	definite article for neutral nouns
noun	*nnum*	the number of the noun	SG	singular
			PL	plural
			MASS	mass noun
			NAME	proper name
noun	*nfrm*	the form of the noun	POS	positive
			DIM	diminutive
			EINF	-e inflection
noun	*nbin*	binding type	NSB	subject bound
			NOB	object bound
adjective	*afrm*	the form of the adjective	NORM	normal
			COMP	comparative
			SUP	superlative
verb	*vfrm*	the form of the verb	INF	infinitive
			PART	particle verb
			PRESP	present participle
			PASSP	passive participle
preposition	*ppos*	the way the preposition must be realized	PREP	preposition
			POST	postposition

Table 3: Overview of parameters, with descriptions of the parameter category (PC) and the parameter value (PV).

3. Parameterized ECM

The central idea behind the parameterized ECM is that many idiom patterns describe structures that are for a large part identical. As shown in Table 1, the description of idiom pattern *IDp1* for idioms such as *de plaat poetsen* is: 'Expressions headed by a verb taking a direct object NP that consists of a determiner and a singular noun.'

In the ECM another idiom pattern such as *IDp2* is required for entries such as *de benen nemen* (lit. 'to take (away) the legs', id. 'to escape'): 'Expressions headed by a verb taking a direct object NP that consists of a determiner and a plural noun.'

And also another idiom pattern such as *IDp3* is required for idioms such as *het loodje leggen* (lit. 'to lay down the piece of lead', id. 'kick the bucket'): 'Expressions headed by a verb taking a direct object NP that consists of a determiner and a diminutive singular noun.'

The only difference between the three idiom patterns is the form of the noun it requires. The use of parameterized equivalence classes reduces the number of idiom patterns, i.e. instead of four different unrelated idiom patterns *IDp1...IDp4*, one might assume a single idiom pattern *IDp5* that takes two arguments (parameters), one to specify the number of the noun, and one to specify whether the diminutive form should be used.

Reducing the number of idiom patterns means reducing the number of equivalence classes. As a result, the number of idioms that have to be dealt with manually minimizes, whereas the number of idioms that can be incorporated into an NLP system in a fully automatic manner increases.

3.1. An overview of parameters for Dutch

In the previous section, I mentioned four potential parameters, viz. singular and plural with respect to the number of the noun, and diminutive and positive with respect to the form of the noun. There is, however, more variation within the individual components of idioms, we can parameterize. In this subsection, I give an overview of the parameters applicable to Dutch idioms.

Recall that the main goal of parameterizing the equivalence classes is to reduce the number of classes, yielding less manual work in the conversion procedure. When determining the aspects we want to parameterize, we must take into account (1) many different frameworks and implementations, and (2) the complexity of the aspect and thus the time we gain with the potential parameter, i.e. each parameter added to the method reduces the number of equivalence classes, but slightly complicates the conversion from the standard representation into a system specific one.

The Rosetta MT system and the Alpino parser[5] were used to examine potential parameters. Both Rosetta and Alpino are Dutch NLP systems. Rosetta is the result of seven years of research on machine translation started in 1985 at the Philips Research Laboratories in Eindhoven. This system is meant to translate between English, Dutch and Spanish and has been developed using compositional translation as guiding principle. The type of grammar used in Rosetta is called *M-grammar*, a computationally feasible variant of *Montague Grammar*.

Alpino is a dependency parser for Dutch, developed in the context of the NWO PIONIER project *Algorithms for Linguistic Processing*. Alpino is based on the Head-Driven

[5] http://www.let.rug.nl/∼vannoord/alp/Alpino/

Expression	ICL
de plaat poetsen (lit. 'to polish the plate', id. 'to clear off')	de plaat[DE][SG][POS] poetsen
de benen nemen (lit. 'to take (away) the legs', id. 'to escape')	de been[HET][PL][POS] nemen
de pijp uitgaan (lit. 'to go out of the pipe', id. 'kick the bucket')	uit[POST] de pijp[DE][SG][POS] gaan
op de fles gaan (lit. 'to go on the bottle', id. 'to go broke')	op[PREP] de fles[DE][SG][POS] gaan
zijn brood verdienen (lit. 'to earn his bread', id. 'make a living')	zijn[DSB] brood[HET][SG][POS] verdienen
iemands hart breken ('break someone's heart')	PNP hart[HET][SG][POS] breken
iemand op handen dragen (lit. 'to carry s.o. on hands' id. 'adore s.o.')	VAR op[PREP] hand[DE][PL][POS] dragen

Table 4: The ICLs of some idioms extended with parameters.

Phrase Structure Grammar (HPSG).

In this approach, the term *parameter* is defined as an occurrence of the pair ⟨parameter category,parameter value⟩, where *parameter category* refers to the aspect we want to parameterize, and *parameter value* to the value a parameter category takes. Table 3 gives an overview of the parameter categories and corresponding parameter values distinguished in this research. Given the parameter categories and parameter values, parameters such as ⟨nnum,SG⟩, ⟨nnum,PL⟩, and ⟨afrm,SUP⟩ can be formed.

3.2. Representation of parameters

The use of parameters in this approach becomes visible in the Idiom Component List (ICL) of the lexical entry of the idiom. Each idiom component in the ICL is represented in the *canonical form*. Since each parameter value is unique, i.e. belongs to only one parameter category, we only represent the parameter value of each parameter. The parameter values are realized between square brackets directly on the right of the item they parameterize. Because of their uniqueness, there is no restriction on the order of values attached to the noun. In the ICL, a preposition-component always precedes its complement, even if it must be realized as a postposition. Determiners are represented with the form they take in the idiom, e.g. the determiner is represented as *de* if the idiom component is a plural definite noun, irrespective of the gender of the noun. In the case of possessive NPs, we use the variable PNP, and VAR is used for obligatory free arguments.

In Table 4 we find some examples of ICLs with parameters. It must be noted that these examples do not necessarily occur in the same equivalence class.

4. Evaluation

Extending the ECM with parameters contributes to reducing the number of equivalence classes and increasing the number of members within each equivalence class. As a result the number of idioms that have to be dealt with manually decreases, whereas the number of idioms that can be incorporated into an NLP system in a fully automatic manner increases. Since the method proposed here categorizes idioms into equivalence classes, the successfulness of the method depends on (1) how many different equivalence classes are distinguished (the less the better), and (2) how many instances each equivalence class contains (the more the better).[6]

In order to determine the effectiveness of the method, I carried out measurements on a database of Dutch idioms. The source I used is an electronic version of the Van Dale Idiom dictionary for Dutch (de Groot, 1999) that contains approximately 6,300 multiword expressions, which includes, besides idioms, also a small number of proverbs and collocations.

The measurements include only three- and four-word idioms. Since the expressions were not grouped according to their type, all three- and four-word expressions – a total of 2,835 – were extracted from the source.

Next, the Alpino parser was used to assign a part-of-speech tag to the components of each expression. The patterns of the idioms, i.e. the equivalence classes, were determined using these part-of-speech tags. For the purpose of this paper, the equivalence classes without a verb were further ignored.[7] All classes with a verb were manually checked for part-of-speech errors, and expressions with a finite verb in canonical form (usually proverbs) were filtered out.

Besides basic part-of-speech tags, Alpino specified other properties, such as the number of the noun, the form of the adjective, etc. This information was used to semi-automatically determine the parameters for each component within an idiom.

In order to measure the number of equivalence classes without parameters, I counted the number of unique parameter combinations from each parameterized equivalence class. For example, in the parameterized ECM the ICL of *de plaat*

[6] The successfulness of the method also depends on the complexity of the incorporation of a parameter into a specific system, which varies from system to system. This point is further addressed in Section 5.

[7] With the risk that idioms do occur in these classes, due to errors in the automatic tagging.

Cov.	# idioms	# ECs	# parameterized ECs
50%	584	29	2
60%	700	45	3
70%	817	67	4
80%	934	102	7
85%	992	132	8
90%	1,050	179	11
95%	1,109	237	15
100%	1,167	295	38

Table 5: Coverage of equivalence classes (ECs).

poetsen (de plaat[DE][SG]POS] poetsen) en de ICL of *de benen nemen* (de been[HET][PL][POS] nemen) occur in the same equivalence class. In the orignal ECM, these ICLs would appear in different equivalence classes, due to the variation of the number of the noun. Each parameter, each unique determiner, and the variable VAR were taken into account.[8]

Table 5 shows the major finding of the measurements. The first row, for example, means that 50% (or 584) of the three- and four-word Dutch idioms considered in this paper can be dealt with by 29 equivalence classes in the original ECM and just two classes in the parameterized ECM. Two main conclusions can be drawn from the results. First of all, introducing parameters in the ECM reduces the number of equivalence classes with almost 90%, and increases the mean cardinality of the equivalence classes with 26,75 idioms assuming a 100% coverage. Secondly, 95% (or 1,109) of the idioms can be dealt with by just 15 parameterized equivalence classes.

Although the evaluation only included three- and four-word idioms, and a considerable rise of equivalence classes is expected when including five- and more-word idioms, the results are promising.

5. Conclusion

In this paper I outlined the problem with MWEs in NLP systems and discussed a very concrete method for a standard for the lexical representation for Dutch idioms originally proposed by Odijk (2003), viz. the Equivalence Class Method. This proposed standard is very simple from a technical and linguistic point of view, it is highly theory-neutral, and it can be an important technique to allow for maximal reuse of lexical entries for idioms in many systems that differ widely in terms of their theoretical basis, their actual implementation, and their treatment of idioms.

In the main part of this paper, I elaborated on one enhancement of the ECM, viz. parameterized equivalence classes. By introducing various parameters suitable for Dutch, I capture in a stringent way relevant generalisations concerning alternations in the idiom structure. As was shown in the evaluation, the introduction of parameters decreases the number of equivalence classes needed with almost 90% with respect to the numbers of equivalence classes needed in the original ECM. A total of 15 parameterized equivalence classes are needed to cover 95% (or 1,109) of the three- and four-word idioms. Concretely, this means that the use of parameters reduces the number of equivalence classes and increases the number of idioms in each class, supporting the task of converting the standard format into the structure required in the target NLP system.

The ability to handle the parameters introduced in this paper varies from system to system. This means that some systems will profit more from the parameterized ECM than other systems. Applications that cannot deal with certain parameters are not harmed, since the original equivalence classes can still be identified.

It must be noted that the purpose this of paper was not to discuss the treatment of idioms in any grammar. In general, adaptions to the grammar must be made in order to incorporate all equivalence classes. However, when a system is able to treat all sorts of idioms in a satisfying way, the proposed method offers a way to incorporate a large number of idioms in the target system with relatively little effort.

6. Future Work

The method as proposed in this paper is still under development. A brief analysis of five-word idioms learns that grouping idioms according to the part-of-speech of the individual components is not always sufficient. What we need is a unique identification of the idiom pattern using a combination of part-of-speech and labels that denote the relation between the individual components. No concrete notation for this relation has been generated yet.

What is also left for future research is to extend the method to larger sets of data and other types of MWEs. Furthermore, the method will be tested in at least two dutch NLP systems, viz. the Rosetta MT system and Alpino.

In this paper *semantic decomposability*, i.e. whether the meaning of the parts of an idiom can be distributed over its parts, was ignored. This is far from desirable, since decomposable idioms are syntactically more flexible than non-decomposable idioms and may require a different treatment in the grammar. Taking into account this distinction may lead to more equivalence classes, or at least more manual work. Mapping the boundaries of flexibility, however, is not always easy and no one can predict exactly which types

[8] In general, the procedure to convert the definite article *the* does not differ from the conversion of the indefinite article *a*. However, in some NLP systems, e.g. the Rosetta system, articles are introduced syncategorematically. This means that a different rule is used for definite articles than for indefinite articles, yielding a different conversion procedure.

of syntactic variation a given idiom can undergo (Sag et al., 2001). More research is needed to give a sophisticated analyses of the syntactic flexibility of Dutch idioms.

7. Acknowledgements

This research is carried out as part of the IRME project, financed by STEVIN (http://taalunieversum.org/stevin). The data were kindly provided by Van Dale Lexicografie.

8. References

Ann Copestake, Fabre Lambeau, Aline Villavicencio, Francis Bond, Timothy Baldwin, Ivan Sag, and Dan Flickinger. 2002. Multiword expressions: Linguistic precision and reusability. In *Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1941–7, Las Palmas, Canary Islands.

Hans de Groot. 1999. *Van Dale Idioomwoordenboek*. Van Dale Lexicografie, Utrecht.

M. Everaert, E-J. van der Linden, A. Schenk, and R. Schreuder, editors. 1995. *Idioms: Structural and Psychological Perspectives*. Lawrence Erlbaum Associates, Hove, UK.

Geoffrey Nunberg, Ivan Sag, and Tom Wasow. 1994. Idioms. *Language*, 70:491–538.

Jan Odijk. 2003. Towards a standard for multi-word expressions. ISLE Project Report, February.

Jan Odijk. 2004a. A proposed standard for the lexical representation of idioms. In *EURALEX 2004 Proceedings*, pages 153–164. Université de Bretagne Sud, July.

Jan Odijk. 2004b. Multiword expressions in NLP. Course presentation, LOT Summerschool, Utrecht, July.

M.T. Rosetta. 1994. *Compositional Translation*. Kluwer Academic Publishers, Dordrecht.

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).