

Evaluation of a machine learning algorithm for MWE identification. Decision Trees.

Begoña Villada Moirón
Alfa-Informatica
University of Groningen
m.b.villada.moiron@rug.nl

October 26, 2006

Abstract

Contents

1	Introduction	2
1.1	MWE identification	2
1.2	Learning the MWE-hood concept	3
1.3	Overview	3
2	Decision trees	4
2.1	Issues that deserve caution	4
2.2	Software	5
2.3	Applied to MWE identification	5
3	Capturing MWE linguistic properties	7
3.1	Measuring lexical affinity	7
3.2	Assessing local context	8
3.3	Morpho-syntactic flexibility	9
3.4	Semantic compositionality	11
4	Modelling and experiments	11
4.1	Candidate data extraction	12
4.2	Reference data	12
4.3	Data annotation and settings	14
4.4	Evaluation methodology	15

5	Results	16
5.1	VERB (NP) PP identification	16
5.2	VERB NP identification	17
5.3	Which features are most useful?	18
5.4	Qualitative evaluation	20
6	Conclusions	26

1 Introduction

1.1 MWE identification

My aim is to identify existing expressions in a large corpus that deserve to be treated as lexical units and therefore, deserve a place in a computational lexicon. Calzolari et al. [2002] proposed the following general definition of MWEs:

Expressions whose linguistic behavior is not predictable from the linguistic behavior of its component words when they occur in isolation.

The set of expressions referred to by this general definition have been named as idioms, institutionalized phrases, support verb constructions, phrasal verbs, compounds (adjective noun or noun noun), collocations (grammatical collocations or lexical collocations), metaphors, etc. We believe that all these subtypes of expressions belong in a lexicon of MWE.

To be able to put our finger down on expressions that behave as a MWE, those expressions that satisfy one or more of the following criteria are considered as MWE:

1. A strong lexical affinity between component words.
2. Limited or non-productive morphology.
3. Limited or restricted morpho-syntactic flexibility.
4. Partially or non-fully compositional meaning.
5. Statistically, an idiosyncratic behavior.

How can we automatically extract the collection of MWEs present in a corpus?

One approach is to devise a probabilistic model that measures for each candidate expression the degree to which one or more of the above criteria is satisfied. Expressions that satisfy more of the above criteria are assigned a

high score, whereas expressions that hardly satisfy any criteria are assigned a low score. The probabilistic model will return a ranked list of candidate expressions in which the top scores are expected to be actual MWES and the low scores are expected to be non-MWES. Stevenson et al. [2004] argue that the lightness of a light verb construction might be best modelled as a continuum as opposed to a discrete distinction. The authors design a statistical measure to identify v NP light verb constructions. In a similar fashion, Fazly and Stevenson [2006] investigate a linear model combining a measure of lexical fixedness and one of syntactic flexibility to extract idiomatic VP constructions from the BNC; their method reaches 80% accuracy.

An alternative approach is to design an identification model as a binary classifier. The classifier goes through a list of candidate expressions splitting them apart into two groups: MWES and non-MWES. To be able to decide which class to assign, the classifier assesses the above list of criteria for each expression, afterwards, the classifier decides on one class. Tan et al. [2006] apply this approach to extract v NP light verb constructions from the parsed Wall Street Journal section of the Penn Treebank. Using association measures as learning attributes, Pecina and Schlesinger [2006] also model extraction of Czech collocations from a treebank as a binary classification task achieving high precision and recall scores.

Both approaches attempt to model the phenomenon of MWES, so to say the underlying concept of MWE-hood present in all MWES.

1.2 Learning the MWE-hood concept

We want to learn the concept of MWE-hood. Among the various types of learning approaches, we choose inductive learning. By using inductive learning, we aim at learning by example. How is the concept learnt? The learner is presented with a collection of labeled examples (training data). All training examples state to which class they are assigned. From observing the training data, the learner builds up hypotheses about what sort of properties are observed in one class and what sort of properties are observed in the other class. After seeing all training data, the learner selects an optimal hypothesis, one that can reliably approximate the target concept.

1.3 Overview

Section 2 provides a brief description of decision trees, how decision trees are used in a classification task, drawbacks and issues one should be aware of and, how the task of MWE identification is envisaged as a classification task using decision trees. Section 3 describes the features used to capture the linguistic properties that presumably distinguish MWES from productive word combinations. Section 4 describes the candidate extraction, data annotation and the lexical resources used, experimental settings and, the

evaluation methodology. Section 5 reports on the quantitative results and qualitative evaluation of the classifiers. Briefly, the findings are summarized in section 6.

2 Decision trees

During training, a decision tree learning algorithm searches for the hypothesis that best approximates the target concept. This hypothesis is formalized as a decision tree.

The target concept is a disjunction of tests applied on the observed features of the training data instances. The disjunction of tests can be read off from the decision tree given that, each internal node represents a test on one of the properties (or attributes) and the node's branches are the possible values of that attribute. The decision tree incorporates all the necessary tests on attributes and their values that are induced from the training data and that are necessary to arrive at an optimal¹ classification of the training data.

During training the algorithm establishes the order in which the tests on the specified properties (attributes) should be applied. First, the algorithm selects an attribute that becomes the root node for the decision tree. The selected attribute is the one with the highest **information gain**, that is, the attribute that can provide the best classification of the training instances. The information gain measures the expected reduction in entropy during the classification. The other attributes are recursively checked according to the information gain criterion and assigned to lower nodes in the tree. Learning stops when all training instances have been checked and the algorithm established an optimal decision tree to classify the training instances.

2.1 Issues that deserve caution

Mitchell [1997] mentions a number of issues to be considered when applying decision trees to a classification task.

Avoid overfitting the data Overfitting arises when the learner learns a target function that classifies the training data with a high accuracy but its performance decreases substantially on unseen instances. Two common situations that may cause overfitting are: (i) random noise in training examples and (ii) if a few attributes seem to partition the data well with a few number of instances under their corresponding tree nodes. Random noise in training data may affect either attribute values that were badly specified or, the target class if wrongly assigned.

¹Mitchell [1997] mentions that a learning bias of the underlying algorithm is that it might not find the best hypothesis but only an optimal one.

Missing attribute values During a classification task, we could face the situation where for a given property we lack the value for one or more instances in the training data. Instead of neglecting the instance altogether, one could (i) assign the most frequent value in training data for that attribute, (ii) assign the most common value of that attribute in those training examples that have class c or (iii) select a value with the highest probability among the training examples that have a certain class c .

Handling numeric attributes Dependency trees perform rather well with attributes that show discrete values. With a discrete-value attribute, all the data is split at the point in the tree where the nominal attribute is tested. The tree uses in one node all the information that the attribute offers. With a numeric-value attribute the values are split into two groups. Later, successive splits on this numeric attribute may continue to yield new information. As consequence, the tree becomes very messy and it is difficult to interpret because the tests on one numeric attribute may be scattered across the tree.

To avoid overfitting the training data, a successful technique is to allow the decision tree to overfit the training data but apply post-pruning of the tree. The software we use (see next section), a C4.5 implementation, applies rule post-pruning. Another alternative is to use a different measure for selecting attributes; a recommended measure is the *gain ratio* proposed by Quinlan (1986).

Training examples that exhibit missing values for one or more attributes are handled according to the third technique (iii) described above. Filling in missing values is done by the C4.5 implementation I use (`weka.classifiers.trees.j48`).

Finally, to avoid splitting numeric attribute checks, one could discretize each numeric attribute, however it is not guaranteed that this would improve the performance enormously. For the time being, I ignore this issue.

2.2 Software

I use a decision tree algorithm available in the freely available WEKA machine learning toolkit [H. and Frank, 2005]. I use the decision trees classifier known as `weka.classifiers.trees.j48`; this classifier implements the C4.5 decision tree learning algorithm.

2.3 Applied to MWE identification

Training data consists of a collection of candidate expressions among which some are MWEs and some are productive word combinations. Table 1 shows a small training data sample. By assessing the training data (labelled examples), the learner should induce an evaluation function that classifies the

example	property ₁ ,property ₂ ,property ₃	class
1	high, low, low,	MWE
2	medium, medium, high,	non-MWE
3	high, nul, low,	MWE
4	low, high, high,	non-MWE
5	high, medium, low,	MWE

Table 1: Collection of 5 labeled examples showing specifications of three properties and the class (MWE is a multiword expression and non-MWE is a regular word combination).

tests	assign class MWE
if (p ₁ =high and p ₂ =low and p ₃ =low)	yes
if (p ₁ =high and p ₂ =low and p ₃ =low)	yes
if (p ₁ =high and p ₂ =medium and p ₃ =low)	yes
if (p ₁ =medium and p ₂ =medium and p ₃ =high)	no
if (p ₁ =low and p ₂ =high and p ₃ =high)	no

Table 2: If-then conditionals learned from the training data displayed in Table 1 above.

training instances correctly. The function learnt from training data can be used again to classify unseen examples. Eventually, we aim at having an evaluation function (or model) that can make accurate predictions on unseen data, i.e. we want to find a model that can classify unseen candidate expressions into MWE or non-MWE with a reasonable accuracy.

A number of features (attributes) have been selected to describe the candidate expressions. These attributes are approximations of linguistic properties that are described in section 3. Before applying the machine learning techniques we ignore which attributes are most useful for MWE identification. Thus, using decision tree learning we also want to establish whether attributes are interesting and which ones are most relevant for the identification task.

The target concept we aim at is MWE-hood. Depending on the property tests an instance belongs to the class MWE ('yes') or it is a productive word combination ('no'). From our little example in table 1, the learner induces possible hypotheses that can best approximate the target concept expressed as disjunctions of 'if-then' conditionals (shown in table 2).

After training is completed, we have

- learnt a model that (on the basis of our training data) approximates the concept of MWE-hood and,
- knowledge about which features are more effective in splitting apart MWEs from productive word combinations.

The decision tree classifier is successful if the model we learnt can classify unseen candidate expressions with a high precision and high recall. In addition, the model should reach a good recall in identifying MWEs. To assess the classification accuracy, the resulting model is tested on a set of unseen candidate expressions (test data).

3 Capturing MWE linguistic properties

I assume that an expression is considered a MWE if it satisfies one or more of the following criteria:

1. A strong lexical affinity between component words.
2. Limited or non-productive morphology.
3. Limited or restricted morpho-syntactic flexibility.
4. Partially or non-fully compositional meaning.
5. Statistically, an idiosyncratic behavior.

Given that I aim at an automated model for MWE identification, the above criteria are approximated with various quantitative measurements calculated on the basis of the contexts in which the MWE candidates or their component words are used in a corpus.

The approximations of the linguistic properties are split into four groups:

1. lexical affinity,
2. local context,
3. morpho-syntactic flexibility and,
4. semantic compositionality.

Next, I describe the quantitative measurements that attempt to assess each of these characteristics.

3.1 Measuring lexical affinity

Co-occurrence frequency gives the number of times the expression ($c(w_1, w_2, w_3)$) has been observed in the extraction corpus.

Words that are seen together very frequently do not necessarily make up a multiword expression (*woon in Nederland*). However, after performing some pre-processing and data filtering one could arrive at a list of word combinations that co-occur a lot and are good examples of MWEs.

Salience and/or log-likelihood A measure that compares the likelihood of seeing the word combination to the likelihood of seeing the combination if its individual components were independent of each other is more informative and therefore, more likely to return a list of idiosyncratic word combinations than basic co-occurrence frequency. The log-likelihood score [Dunning, 1993] and the salience [Kilgarriff and Tugwell, 2001] measures are examples of such a measure. Salience, a variant of pointwise mutual information, has been shown to give satisfactory results in identifying collocations in lexicographic projects and support verb constructions [Villada Moirón, 2005]. In my preliminary experiments, salience is more useful scoring technique than the log-likelihood, thus, I chose to use salience as attribute.

Support verb Support verb constructions make an important group among MWES. Although identifying support verb constructions is not trivial, for an automated identification model it might be beneficial to know that certain verbs in the language may have a use as a support verb. Concerning Dutch, nine verbs that can function as main but also as light verbs are *brengen* ‘bring’, *doen* ‘do’, *gaan* ‘go’, *geven* ‘give’, *hebben* ‘have’, *komen* ‘come’, *krijgen* ‘get’, *maken* ‘make’, *nemen* ‘take’ and *stellen* ‘state’ [Hollebrandse, 1993]. To this list I added *houden* ‘hold’.

Thus, if the verb in a candidate expression can also be used as a support verb, the candidate specifies a binary feature **support** with value **y(es)** or **n(o)**.

3.2 Assessing local context

Head dependence Merlo and Leybold [2001] use the head dependence as a diagnostic to determine the argument status (or not) of a PP. PP-arguments should only appear with a group of head verbs that lexically select them, while PP-modifiers can appear with a greater range of different head verbs.

Head dependence measures the size of the set of verbs that select for a given PP. Baldwin [2005] also suggests that this measure ought to be helpful to identify prepositional verbs. In order to quantify the head dependence, two techniques proposed by Merlo and Leybold [2001] are used:

- the number of verbs that select a given PP: *hd_d_int* and,
- the entropy of the distribution among the verbs that select for a given PP (represented as a (P, N) tuple) given in equation 1

$$H(P, N) = - \sum_j \frac{f((P, N), V_j)}{f(P, N)} \log \frac{f((P, N), V_j)}{f(P, N)} \quad (1)$$

I also use head dependence for v NP identification. To compute the relevant scores, I substitute the PN tuple by the NOUN in the above equation.

Dependency relation and frequency The Alpino parser assigns a dependency relation to every dependent of a verb, e.g. direct object (**obj1**), prepositional complement (**pc**), separable particle or fixed phrase (**svp**), modifier (**mod**), etc. This dependency relation may provide useful information about what sort of relationship holds between a verb and a PP in a candidate expression. A PP such as *in de stad* might always be assigned the dependency relation **mod**, whereas an argument PP such as *aan de pols* (in *vinger aan de pols houden*) required by a verb might most of the time be assigned a **pc** dependency relation. This could be useful information to detect how strongly associated are a verb and a dependent phrase (PP).

In v (NP) PP identification, we use:

- The dependency relation assigned by the parser to the PP inside a candidate expression; for each candidate, we select the dependency relation with the highest relative frequency observed across all the instances of the candidate.
- The relative frequency of the most likely dependency relation.

3.3 Morpho-syntactic flexibility

Preference for a specific syntactic context In verbal predicates there are obligatory and optional syntactic dependents. Among obligatory dependents there are productive complements, fixed arguments in MWES and predicative phrases. Among optional ones, there are adjunct modifiers. Fuzzy cases exist which are hard to classify.

Broekhuis [2004] studied the necessary and sufficient conditions to label a PP as a complement or as an adjunct. Among the syntactic tests to distinguish complements from adverbial PPs (in Dutch), [Broekhuis, 2004, pg.107] observed that the PP complements are closer to the verb in verb final context than adverbial PPs. A verb final context is found in Dutch subordinate clauses. In such context, modals, auxiliary and main verbs form the verb group that occupies the final position in the clause. Jack Hoeksema (p.c.) further maintains that fixed arguments in fixed expressions show a preference for the position immediately preceding the verb group. We apply these claims hoping that arguments in MWES show a stronger tendency than other regular complements and adjuncts to occur in preverbal position in a subordinate context.

Thus, given an observed VERB PP, we measure the relative frequency of seeing the PP immediately preceding the verb group (shown in equation 2).

It is expected that required arguments in MWEs show a high relative frequency, whereas adjuncts and modifiers show a lower frequency. Regular complements of a main verb might show a high frequency, too.

$$f((V, P, N), pos = 'ipr') = \frac{c((V, P, N), pos = 'ipr')}{c((V, P, N))} \quad (2)$$

The same measure is used to measure how strong is the tendency of seeing an argument NP in a VERB OBJECT MWE immediately preceding the verb group in a verb-final context.

Modifiability I measure the modifiability observed inside the argument PP (or NP) in a potential MWE. I used a heuristic proposed by Wermter and Hahn [2004]. Wermter and Hahn [2004] identify the most characteristic lexical material within a phrase by simply calculating the relative frequency of each PP (or NP) realization (see equation 3). The most characteristic realization is the one with the highest relative frequency. This relative frequency is one of the scores of modifiability.

$$f((V, P, N), det_adj_morph) = \frac{c((V, P, N), det_adj_morph)}{c(V, P, N)} \quad (3)$$

Wermter and Hahn’s modifiability measure tells us how salient a given realization is if compared to all other observed variants of an expression. A low score assigned to the most frequent supplement suggests that the expression allows plenty of variation. A high score assigned to the most frequent supplement suggests that the expression is rather (or totally) fixed.

It is also useful to know whether an expression shows many variants and whether such variants are very frequent or not. I also approximated modifiability with standard entropy. The entropy of an expression is computed as the entropy observed among the realizations of the determiner, adjective and noun inflection slots.

Passivization A syntactically-flexible expression may be encountered in WH-extraction, clefting, topicalization contexts, etc. If the head verb in the expression is transitive, syntactic flexibility can also be manifest as passivization.

This feature records how often a candidate expression has been observed in a passive construction (expressed as a relative frequency).

Pronominalization Referential NP phrases can usually be pronominalized in any language. It is generally agreed that NPs in expressions with a non-compositional meaning cannot be pronominalized. Bearing this in mind, the feature 'pronominalization' records whether the NP object inside a PP is realized by a pronoun or not. Candidate expressions whose PP argument includes a pronoun are unlikely to constitute a true MWE.

3.4 Semantic compositionality

Compositionality of meaning has long been proposed as the characteristic that distinguishes productive word combinations from idiosyncratic expressions such as idioms. An expression with non-compositional meaning deserves lexical mention, however, I do not believe that all expressions that deserve lexical mention exhibit a non-compositional meaning. There are institutionalized phrases and other rather literal MWEs whose meaning looks fully compositional. For those of us aiming at identifying MWEs, if we know that an expression has a partially compositional or a non-compositional meaning is sufficient evidence to labelling the expression as a MWE.

In ongoing work, Tim van de Cruys investigates unsupervised clustering techniques to group nouns and verbs into semantically related classes. Such techniques group lexemes (e.g. nouns, verbs) that exhibit similar selectional preferences; the techniques build on work by Resnik (1993,1996). I only mention the two semantic scores used as features: (i) semantic uniqueness which is approximately a ratio between the selectional preference of a *verb prep* for a *noun* and the selectional preference of a *noun* for a *verb prep* and (ii) the selectional association between a VERB PREP and a NOUN. Full description of these scores will be soon available in (van de Cruys, forthcoming).

4 Modelling and experiments

The models are tested on two different target MWEs. First, I test the identification method on VERB (NP) PP combinations and second, on VERB NP combinations.

Examples of VERB (NP) PP MWEs are:

- (1) *aan bod komen*
- (2) *de vinger aan de pols houden*

We have in mind VERB NP combinations whose verb and object NP (the accusative object) form a meaning unit. Examples of VERB NP MWE are:

- (3) *spijt hebben (van)*
- (4) *een brug slaan*
- (5) *open kaart spelen*

(6) *het hoogste woord voeren*

Among VERB NP combinations, some show a prepositional complement, e.g. *spijt hebben van* and *akte nemen van*. It's unclear what the status of the PP introduced by the preposition is. In any case, the object of the preposition is not fixed but an open variable slot. I limit the exercise to establishing how well the methods are able to identify VERB NP MWES, ignoring for the moment whether the PREP phrase also introduces a complement of the verb or not.

4.1 Candidate data extraction

In these experiments, V (NP) PP and V NP candidates are extracted from the CLEF corpus. CLEF consists of two years of two newspapers, thus, having ca. 80 million words and 4 million sentences. The corpus has been automatically annotated with the Alpino parser. Thus, each sentence in the corpus has been fully parsed.

In V (NP) PP identification, all occurrences of a PP and its selecting verb are collected. We ignore PPs that are not dependents of a verb, that is, PPs that may modify a noun. Table 3 lists all raw features necessary to code the linguistic properties (earlier described in section 3) as attributes for the machine learning learner. In V NP identification, we collect all occurrences of an NP being selected by a verb and tagged as an `obj1` (accusative object).² All features in table 3 except (2) are used in the extraction of V NP.

The distribution of the extracted candidate expressions is shown in Table 4. For the task of VERB (NP) PP identification, 8,488 different types with a frequency equal or greater than 10 are extracted. For the task of VERB NP identification, 10,211 candidate expressions are available.

4.2 Reference data

To evaluate the identification models, data from two large lexical databases is used: the Van Dale VLIS database and the *Referentiebestand Nederlands* RBN [Martin and Maks, 2005].

Vlis database The database consists of more than 61,150 idioms and collocations collected by professional lexicographers. In principle, all these expressions are considered as a multiword expression. Idioms, proverbs, sayings, collocations, etc. are all found. A variety of syntactic patterns are found in this database: VERB OBJECT, VERB (NP) PP, ADJ NOUN, ADJ VERB, etc.

²Verbal predicates from which V NP candidates were extracted may include one (or two) extra obligatory dependent NP, ADJP, PP, etc. As a result, a V NP candidate is extracted while in fact, the actual expression shows a different syntactic pattern. If the candidate is classified as a correct MWE, this candidate is wrongly assigned to the syntactic pattern V NP. This is a limitation to the extraction technique that will be improved in the future.

(1)	verb	verb tense	lexeme finite/infinitive
(2)	PP	head obj NP location dependency relation	preposition head noun wrt to verb
(3)	NP	determiner adjective post-nominal mod number	head POS, lexeme singular/plural
(4)		pronoun	yes/no
(5)	SUBJ	head noun	
(6)	OBJ1	head noun	
(7)	passive	yes/no passive auxiliary	

Table 3: Raw features extracted from parsed data.

Syntactic pattern	Types	Tokens
V PP	4,969	
V NP PP	3,519	
total	8,488	1,140,800
V NP	10,211	2,053,286

Table 4: Candidates distribution across extracted patterns ($freq \geq 10$).

Form	Expression
Canonical	uit zijn dak gaan
Formalized	uit/uit/prep/0 zijn/zijn/det/2 dak/dak/noun/4 gaan/ga/verb/6

Table 5:

Syntactic pattern	Freq	Types	MWES	non-MWES
V (NP) PP	≥ 10	8,488	1,910 (22.5%)	6,578 (77.49%)
V NP	≥ 10	10,211	2,771 (27.13%)	7,440 (72.86%)
	≥ 50	1,769	917 (51.83%)	852 (48.16%)

Table 6: Candidates distribution across extracted patterns.

RBN The RBN is a reference lexical database; it includes singleton lexical entries but also what we consider MWES. These are included within the noun, adjective and adverb lexical entries. They are specified as ‘combinatorics’. Currently, I use a list of 3,805 MWES extracted from the original database. Many of these overlap with the Vlis database.

To be able to compare the output of the identification models to the MWES existing in the two lexical databases, the expressions in the databases were parsed with the Alpino parser. The result is that all expressions are formalized. An example expression, *uit zijn dak gaan* ‘go crazy’ is shown in Table 5. The formalized expression shows word form, Alpino’s root form, part of speech and position in the expression. The motivation to formalize all expressions in the reference data is to ensure that the output of the identification models is assessed against the reference data in an as accurate as possible fashion.

4.3 Data annotation and settings

All candidate expressions were checked automatically against the two lexical resources described in section 4.2. Having annotated the datasets automatically, the distribution of actual MWES and non-MWES is shown in table 6. It is important to notice that in both datasets and when using a low frequency cutoff ($f \geq 10$), the proportion of non-MWES is 3/4 of the data, thus the learner has a lot more evidence about one class of expressions. I expect the classifier to reach better performance on classifying non-MWES. With a higher frequency cutoff ($f \geq 50$), the V NP data is almost uniformly split across the two classes. The classifier should perform equally on both classes.

Given that the extraction method is fully automatic, among the candidate expressions in the V (NP) PP data there might be actual instances of a syntactic pattern other than the one I am interested. Among the candidate

Classifier	Attributes	
	V (NP) PP	V NP
C1	freq, salience, support	freq, salience, support
C2	hd_dep_int,hd_dep_ent, dep_rel, dep_rel_freq	hd_dep_int,hd_dep_ent
C3	rel_freq_pn_vfi, modifiab. passive, pron	rel_freq_pn_vfi, modifiab. passive, pron
C0_1	all features	all features
C0_2	all features + semantic scores	

Table 7: Attributes used by the classifiers.

expressions in the V NP data, I expect to find instances of other syntactic patterns such as NP ADJ V, NP PP V, etc.

In the V (NP) PP identification task, I run different experiments by (i) splitting the data into training and test data and (ii) using all data as training data. Experiments were done with 60%, 50%, 40% of training data and the rest of testing data. In the V NP identification task, I only run experiments using all data as training data. These are still rather preliminary experiments.

In V (NP) PP identification, I compare the five classifiers specified in table 7 with the features shown in the middle column. In V NP identification, I only compare 4 classifiers with the features specified in the right column.

No parameter tuning has been done, thus I applied a base decision trees classifier. Given that most of the attributes used are numeric, I investigated the performance of the classifiers when using other feature selection criteria. Instead of the information gain measure, experiments were done with the gain ratio and a χ^2 measure. The results of applying these measures did not differ much, thus I do not include them in the presentation.

4.4 Evaluation methodology

To assess the performance of the classifiers, these are compared to a baseline classifier. In all experiments, baseline reports on a naive classifier that always chooses the most frequent class. The baseline for each dataset can be read off from table 6. Each classifier was first trained on 60% of the labeled data and tested on 40% of labeled unseen data. In addition, the classifier was tested on 'All' data. For evaluation, the performance of the classifier on training data is given when applying 10-fold cross-validation. The performance of the classifier on test data is plain accuracy.

Accuracy gives the proportion of candidates that were correctly classified. Thus, accuracy reflects how good is the classifier in assigning a candidate

Classif	Dataset	Accuracy	class 'y'			class 'n'		
			P	R	F	P	R	F
C1	Training (10fcv)	81.46	0.62	0.43	0.51	0.85	0.92	0.88
	Test	81.00	0.62	0.43	0.5	0.84	0.92	0.88
	All (10fcv)	81.22	0.64	0.36	0.46	0.83	0.94	0.88
C2	Training (10fcv)	80.55	0.6	0.37	0.46	0.83	0.93	0.88
	Test	80.93	0.61	0.44	0.51	0.84	0.91	0.88
	All (10fcv)	81.52	0.62	0.43	0.51	0.85	0.92	0.88
C3	Training (10fcv)	80.02	0.62	0.27	0.37	0.81	0.95	0.88
	Test	81.29	0.7	0.31	0.43	0.82	0.96	0.88
	All (10fcv)	80.53	0.66	0.26	0.38	0.81	0.96	0.88
C0_1	Training (10fcv)	82.52	0.64	0.5	0.57	0.86	0.91	0.89
	Test	82.07	0.62	0.47	0.54	0.86	0.91	0.89
	All (10fcv)	82.99	0.67	0.483	0.561	0.86	0.93	0.89
C0_2	Training (10fcv)	82.71	0.65	0.5	0.56	0.86	0.92	0.89
	Test	82.75	0.64	0.49	0.56	0.86	0.92	0.89
	All (10fcv)	83.4	0.66	0.53	0.59	0.87	0.92	0.89
Baseline	All	77.49	0	0	0	1.0	1.0	1.0

Table 8: V (NP) PP experiment results (frequency ≥ 10).

expression to one class or another. Two of the datasets show a rather skewed distribution of MWES and non-MWES (even though probably very realistic). A naive classifier that always chooses the most frequent class would, without effort, reach a high accuracy. Unfortunately, such high accuracy would not reflect a clever performance of the classifier.

Precision, Recall and F-measure The results also specify precision (P), recall (R) and the F-measure (F) per class, that is, for MWES (class 'y') and non-MWES (class 'n'). These 'per class' evaluation measures are illustrative of how useful the classifier can be to identify MWES in unseen data, whether the precision is sufficient and whether the classifier recall is optimal.

5 Results

5.1 VERB (NP) PP identification

Table 8 shows the performance of all classifiers (C1, C2, C3, C0_1, C0_2) in the task of VERB (NP) PP identification. The overall accuracy of the classifiers on the 'All' dataset (10-fold cross validation) shows improvement over the baseline. In addition, this accuracy suggests that the difference in performance between the single classifiers (assessing lexical affinity, local context

Classif	Dataset	Accuracy	class 'y'			class 'n'		
			P	R	F	P	R	F
C1	All (10fcv)	76.81	0.66	0.29	0.4	0.78	0.94	0.85
C2	All (10fcv)	73.52	0.54	0.13	0.22	0.74	0.95	0.84
C3	All (10fcv)	74.05	0.6	0.13	0.21	0.74	0.96	0.84
C0	All (10fcv)	77.06	0.61	0.41	0.49	0.8	0.9	0.85
Baseline	All	72.86	0	0	0	1	1	1

Table 9: Experiment results on verb object NP (Frequency ≥ 10).

or morpho-syntactic flexibility) is not substantial. A superior classifier that combines lexical affinity, local context and morpho-syntactic flexibility information shows a slight improvement; furthermore, there is accuracy to be gained by adding semantic information.

With a data split into 60% training and 40% testing, the accuracy of the classifiers is fairly stable. This suggests that the classifiers do not seem to overfit the data.

The performance of the classifiers varies a lot 'per class'. The classifiers show higher precision, recall and f-measure scores on non-MWES than on MWES. This was expected given that the classifiers have 3 times more evidence about the negative expressions. The performance of the classifiers within a class does not show significant differences. However, the best precision, recall and f-measure scores are obtained with the superior classifier (using lexical affinity, local context, morpho-syntactic flexibility and semantic information).

5.2 VERB NP identification

Table 9 and table 10 show the results of applying 4 classifiers on V NP data with a frequency cutoff $f \geq 10$ and $f \geq 50$, respectively. Recall from table 7 that the attributes used within these classifiers diverge slightly from those used in V (NP) PP identification. No semantic scores were used in these experiments but they will be added in future experiments.

In both datasets, the overall accuracy is better than the baseline, therefore, the informed classifiers achieve a better accuracy. Caution should be taken, though. The accuracy rise still allows plenty of room for improvement.

On the dataset that includes more low frequency (ref. Table 9), the classifier assessing only lexical affinity (C1) reaches almost the same accuracy as the superior classifier (C0); On the more uniformly split dataset (ref. Table 10), the classifier assessing only the local context (C2) reaches the same accuracy as the superior classifier (C0). However, the difference in accuracy is so small that it is difficult to draw any conclusions from this.

Classif	Dataset	Accuracy	class 'y'			class 'n'		
			P	R	F	P	R	F
C1	All (10fcv)	63.03	0.66	0.58	0.62	0.6	0.68	0.63
C2	All (10fcv)	65.34	0.62	0.8	0.7	0.7	0.48	0.57
C3	All (10fcv)	64.33	0.65	0.67	0.66	0.63	0.61	0.62
C0	All (10fcv)	65.29	0.65	0.68	0.67	0.64	0.61	0.63
Baseline	All	51.83	1	1	1	0	0	0

Table 10: Experiment results on verb object NP (Frequency ≥ 50).

Examination of precision, recall and the f-measure scores re-confirm our earlier claims related to the V (NP) PP dataset: the classifiers perform better on the non-MWE class given that there is more available evidence. If we compare these three scores in the dataset which is uniformly split into MWES and non-MWES, the 'per-class' P, R, F scores are closer to one another. It is important to mention that

- the precision within the MWES class is rather stable across the frequency spectrum;
- when the dataset includes low-frequency data, recall of MWES is very poor.

It is known that many MWES occur with a low-frequency in corpora. In fact, more than twice as many true MWES are present in the dataset that includes low-frequency data than in the dataset with a higher frequency cutoff (ref. to table 6). A successful identification method should also reach a reasonable recall with low-frequency data. This preliminary results suggest that give a bigger extraction corpus and more training data, much better precision and recall may be reached.

We refrain from making any further observations given that these results (concerning VERB NP identification) are still very preliminary.

5.3 Which features are most useful?

WEKA can perform evaluation of the information gain supplied by each feature during classification. If we rank the features according to their information gain we get the ranked feature scale shown in (4). It is remarkable to find that the dependency relation has the highest information gain. Preference for a specific syntactic context is more informative than salience, head dependence, modifiability and the semantic scores. I suspect this ranking is a result of applying a superior classifier that can reach a reasonable accuracy by getting the most frequent class ('no') right.

Features used	Accuracy	P	R	F
All	83.4	0.66	0.53	0.59
- pron	83.4	0.66	0.53	0.59
- supp	83.57	0.68	0.51	0.58
- passive	83.84	0.67	0.49	0.57
- dep_rel_freq	83.42	0.68	0.48	0.57
- s1	83.02	0.67	0.47	0.55
- s2	82.84	0.66	0.47	0.55
- modif	83.04	0.7	0.42	0.52
- hd_int	82.96	0.7	0.42	0.52
- freq	82.56	0.64	0.49	0.55
- hd_ent	82.42	0.65	0.47	0.54
- salience	80.8	0.78	0.2	0.32
- rel_freq_pn_vfi	80.82	0.78	0.2	0.32
Baseline	77.49	0	0	0

Table 11: Accuracy, precision (P), recall (R), f-measure (F) resulting from sequentially removing individual features from the superior classifier C0_2. The last three scores refer to the class 'y'.

$$\begin{aligned}
& \text{dep_rel} \prec \text{rel_freq_pn_vfi} \prec \text{salience} \prec \text{hd_ent} \\
& \quad \prec \text{freq} \prec \text{hd_int} \prec \text{modif} \prec \text{s2} \prec \text{s1} \\
& \quad \prec \text{dep_rel_freq} \prec \text{passive} \prec \text{supp} \prec \text{pron}
\end{aligned} \tag{4}$$

What is the effect of each individual feature in the overall performance of the classifier? Table 11 provides the accuracy, precision (P), recall (R), f-measure (F) resulting from sequentially removing individual features from the superior classifier C0_2. The last three scores refer to the class 'y'. Reading downwards along a column, the scores reflect the effect of removing one feature at a time from the superior C0_2 classifier. Reading upwards along a column, the scores reflect how adding an extra feature the overall accuracy slightly improves. Worth mentioning is the fact that the most important effect of adding features that approximate linguistic properties is to gain in recall, i.e. find a classifier that while being relatively precise in identifying MWEs, it also extracts as many of the MWEs existing in a corpus as possible.

On the basis of the above scores, we examined how do precision and recall change after the addition of extra features. Head dependence (entropy), salience, modifiability and the semantic scores have a negative effect on precision but a positive effect on recall. On the other hand, frequency has a positive effect on precision but a negative effect on recall. These observations

follow from our experimental conditions but we ignore whether they are applicable in general.

5.4 Qualitative evaluation

We inspect the classification results of the v (NP) PP candidates to establish (i) whether some generalizations about the features used can be drawn from inspecting the correctly classified MWES and (ii) what errors the best classifier make.

CORRECTLY CLASSIFIED CANDIDATES

Table 12 shows example candidates that are correctly classified as MWES by the two superior classifiers C0.1 and C0.2. High and low frequency true positives are shown.

Some facts observed among candidates that were correctly classified as MWES call our attention. All frequency ranges are found, some show the minimum allowed frequency ($f = 10$). Saliency scores range from 108.42 (*nieuw leven inblazen*) until -7.66 (*in het leven komen*). There is no apparent threshold one can establish above which we have MWES. More correctly classified expressions exhibit a verb that may not be used as support (697 vs. 592). The head dependence measured with entropy ranges from 0 (*ten prooi vallen*) to 5.41 (*aan eind komen*). The trend observed is that true MWES exhibit low head dependence scores. Concerning the dependency relation observed in the argument PP, this PP can be assigned an **ld**, **svp**, **pc**, **mod**, **predc** or **predm** dependency relation.³ Argument PPs in this type of MWES show a high relative frequency of occurring in a position immediately preceding its selecting verb within the verbal cluster (≥ 0.81 , with a few expressions showing a lower frequency). The modifiability of the argument PP varies a lot, with those expressions that are more idiomatic showing no variation (e.g. *iem aan de tand voelen*, *iets uit het oog verliezen*) and expressions that are fairly transparent and compositional showing substantial variation (e.g. *tot conclusie komen*, *op een idee brengen*). Furthermore, knowing that all instances of an expression are in passive contexts or not does not mean anything to decide on the expressions MWE-hood. No obvious patterns emerge in the behavior of the semantic scores within the candidates classified as MWES. Known to us is that adding the semantic scores increases the number of correctly classified MWES as shown in Table 13.

A few annotation errors surface among expressions correctly classified as MWES: *ga bij mij*, *maak van er*, *slaag np in er*. These are introduced during automatic annotation of the candidate expressions.

³This is interesting. If we were to annotate these MWES in a computational grammar, one should consistently annotate the dependency relation of the argument PP. Nonetheless, it may be the case that lexicon developers consider some of the v (NP) PP MWES correctly identified as a combination of a VERB and a PREDICATIVE PP.

triple	MWE	freq
kom#nul#om#leven	<i>om het leven komen</i>	2705
kom#nul#tot#stand	<i>tot stand komen</i>	2005
neem#np#in#beslag	<i>iets in beslag nemen</i>	1723
kom#nul#op#gang	<i>op gang komen</i>	1433
kom#nul#in#actie	<i>in actie komen</i>	1303
heb#np#in#handen	<i>iets in handen hebben</i>	1132
kom#nul#ten#goede	<i>iets ten goede komen</i>	1132
kom#nul#aan#orde	<i>aan de orde komen</i>	1096
heb#np#achter#rug	<i>iets achter de rug hebben</i>	1044
geef#nul#de#voorkeur	<i>de voorkeur geven</i>	1032
kom#nul#aan#bod	<i>aan bod komen</i>	1032
neem#np#voor#rekening	<i>iets voor zijn rekening nemen</i>	1030
ben#nul#van#plan	<i>van plan zijn om</i>	988
kom#nul#te#pas	<i>aan/bij/in iets te pas komen</i>	983
breng#np#op#markt	<i>iets op de markt brengen</i>	975
ga#nul#ten#koste	<i>ten koste gaan</i>	951
ga#nul#van#start	<i>van start gaan</i>	886
houd#np#in#gaten	<i>iets in de gaten houden</i>	831
ga#nul#in#beroep	<i>in beroep gaan</i>	803
kom#nul#tot#conclusie	<i>tot de conclusie komen dat</i>	755
ga#nul#ten#onder	<i>ten onder gaan</i>	744
kom#nul#aan#licht	<i>aan het licht komen</i>	719
heb#np#in#huis	<i>iem.iets in huis hebben</i>	699
heb#nul#tot#gevolg	<i>iets tot gevolg hebben</i>	693
kom#nul#ter#sprake	<i>ter sprake komen</i>	640
hink#nul#op#gedachte	<i>op twee gedachten hinken</i>	10
houd#np#buiten#schot	<i>zich buiten schot houden</i>	10
houd#np#in#balans	<i>iets in balans houden</i>	10
houd#np#in#hechtenis	<i>iem. in hechtenis houden</i>	10
leef#nul#in#onmin	<i>in onmin leven met</i>	10
leef#nul#onder#armoede_grens	<i>onder de armoedegrens leven</i>	10
leg#np#te#vondeling	<i>een kind te vondeling leggen</i>	10
neem#np#onder#schot	<i>iem. onder schot nemen</i>	10
neem_op#np#in#inrichting	<i>iem. opnemen in een inrichting</i>	10
ontruk#np#aan#vergetelheid	<i>iets aan de vergetelheid ontrukken</i>	10
overtuig#np#van#er	?	10
prijs#np#uit#markt	?	10
richt#np#op#toekomst	<i>zich richten op de toekomst</i>	10
schreeuw#np#van#dak	<i>iets van de daken schreeuwen</i>	10
sleep#np#uit#vuur	<i>iets uit het vuur slepen</i>	10
stap#np#naar#rechter	<i>naar de rechter stappen</i>	10
steek#nul#naar#kroon	<i>iem. naar de kroon steken</i>	10
stel#np#in#licht	<i>iets in een ... licht stellen</i>	10
stel#nul#onder#curatele	<i>iem onder curatele stellen</i>	10
stort#np#in#fonds	<i>geld storten in een fonds</i>	10
teken_op#np#uit#mond	<i>iets uit iemands mond optekenen</i>	10
verander#np#van#gedachte	<i>van gedachte veranderen</i>	10
verdwij#np#uit#zicht	<i>uit het zicht verdwijnen</i>	10
verkrijg#np#over#hart	<i>iets niet over zijn hart kunnen verkrijgen</i>	10
zet#np#op#zjspoor	<i>iem op een zijspoor zetten</i>	10

Table 12: Correctly classified MWES.

Classifier	MWES	non-MWES
C0_1	1048	6371
C0_2	1290	6318

Table 13: How many expressions are correctly classified?

Classifier	MWES	non-MWES
	fns	fps
C0_1	861	206
C0_2	619	259

Table 14: How many expressions are misclassified?

To summarize, after inspecting the behavior of the features across the correctly classified candidates (true positives), no feature on its own leads to a correct classification. The classifier makes a decision on the basis of a series of sequential checks on each of the attributes available. Given that many of the attributes have a numeric value, more than one check on such attribute is performed (this we can observe in the resulting decision tree).

MISCLASSIFIED CANDIDATES

The superior classifiers (C0_1 and C0_2) misclassify more candidates that are MWES than candidates that are non-MWES. Table 14 shows the number of false negatives and false positives per classifier. *False negatives* are those candidates that exist in our reference data but which the classifier labeled as 'non-MWE'. *False positives* are those expressions classified as MWE but are absent in our reference data. We concentrate on the CO_2 classifier, given that its classification accuracy is slightly better yielding fewer false negatives and more false positives.

Among false negatives,

- about 36% are **annotation errors** introduced during automatic annotation. These candidates were annotated as 'yes' but the classifier proposes a 'no' label. The reasons to consider this 36% as errors are that
 - candidate expression belongs to another syntactic pattern. Candidate `met elkaar gaan` matches *broederlijk met elkaar gaan* (adj pp v); `op elkaar houden` matches *zijn haken stijf op elkaar houden* (np adj pp v)
 - candidate is part of a proverb/saying: `kom met gulden` matches *met drie gulden kom je niet ver*

- candidate is incomplete given that the actual MWES containing two verbs (e.g. laten+infinitive): **zit op zich** could be a true match of *iets niet op zich laten zitten*, **hoor van zich** matching *niets van zich laten horen*, **lig in zon** matching *in de zon liggen braden*
- candidate is part of a construction found in VLIS data: **gaat om principe** which matches *het gaat om het principe/idee/spel/volgende ...*
- about 63% are **classifier errors**
 - most errors are rather literal MWES (*in de file staan, naar de stembus gaan, voor het raam zitten*); metaphors and idiomatic combinations are also found (*door de bocht gaan, naar de pen grijpen, bij moeders pappot zitten, in iemands vaarwater zitten*)
 - triple is ambiguous; it could represent different MWES of the syntactic pattern we aim at. E.g. **heb in hand** could stand for *iets in de hand hebben, iets in eigen hand hebben, iets niet meer in de hand hebben*, etc.
 - a large part of the false negatives show the verbs *komen, hebben* or *zitten*.

It is difficult to see whether patterns or correlations exist across or between the attributes describing the false negatives. One can measure the Spearman’s rank correlation between vectors resulting from ordering the list of false negatives according to each numeric attribute. The correlation coefficients between a pair of two attributes are given in table 15. The coefficients can be useful to establish e.g. whether the value of modifiability correlates with the value of ‘s1’ across all examined false negatives. As shown in row 4, column 4, the coefficient is very small and negative, evoking a negative correlation. Generally speaking, there are no significant correlations between the attributes observed among the false negatives (classifier errors).

Among false positives there are:

- **a_tp**: about 23% are annotation errors but true positives. The classifier proposes ‘yes’ but candidates were incorrectly annotated as ‘no’ during automatic annotation. Plausible reasons are
 - noun form in triple does not match noun in dictionary (**in werk stellen** corresponding to *iets in werking stellen*)
 - verb form in triple does not match verb in dictionary (**geraak in opspraak** not matching *in opspraak raken*)
 - expression is missing in parsed reference data: *iets te grazen nemen*

Attribute	lexaff	ppipr_freq	modif	s1	s2	hd_int	hd_ent
lexaff	-	-	-	-	-	-	-
ppipr_freq	-0.041	-	-	-	-	-	-
modif	0.0096	-0.046	-	-	-	-	-
s1	0.02	-0.136	-0.024	-	-	-	-
s2	0.007	0.007	0.086	0.144	-	-	-
hd_int	0.11	0.084	0.087	0.005	-0.017	-	-
hd_ent	-0.043	0.067	0.085	0.0089	0.015	0.113	-
passive	0.058	-0.147	-0.114	-0.122	-0.2	-0.063	0.044

Table 15: Spearman’s rank correlation coefficient scores between some of the attributes.

- preposition or argument PP shows a spelling variant: *ten gronde gaan* and *zich ten gronde richten* (instead of *te gronde . . .*), *iets te beschikking stellen* (instead of *ter*), *schiet te kort* instead of *tekortschieten*
- **a_sp**: about 14% are annotation errors due to the fact that the candidate matches a MWE in the reference data but this MWE shows another syntactic pattern:
 - *te werk gaan* is part of the MWE ADJ/PP TE WERK GAAN.
 - *in zee gaan* is part of the MWE *in zee gaan met iemand*
- **te**: about 60% are true errors Some of these are shown in table 16.
 - *betrap op gebruik*
 - metaphorical expressions: *in probleem brengen*, *op schroef zet* (op losse schroeven zetten?)
 - transparent expressions *iets in het nieuws komen*
 - predicative PPs with a verb (not in reference data) *onder druk komen*, *aan de macht zijn*, *in voorbereiding hebben*, *in omloop komen*, *aan de kook komen*, *in zwang zijn*, *aan slag zijn*, *onder controle brengen*, *aan leiding komen*
 - grammatical collocations with an open NP object: *van je houden*, *tegen mij zeggen*, *benoemen tot directeur*,
 - directional PPs: *naar school + brengen*, *naar moskee gaan*, *naar de huisarts gaan*
 - locative PPs: *iets op de agenda zetten*
 - determinerless PPs plus verb: *op orde*, *als verklaring*

To summarize,

triple	freq
breng#np#naar#buiten	373
heb#nul#tot#doel	294
kom#nul#in#nieuws	266
zit#nul#op#tribune	220
breng#np#naar#ziekenhuis	193
zit#nul#in#auto	176
breng#np#in#probleem	146
kom#nul#onder#druk	145
zit#nul#in#zaal	140
ga#nul#om#vraag	137
zit#nul#in#hoofd	118
ga#nul#naar#buitenland	108
breng#np#op#voorsprong	83
kom#nul#op#voeten	83
breng#np#op#scherm	73
sta#nul#achter#naam	71
ben#nul#aan#macht	70
neem#np#met#plaats	69
zit#nul#op#fiets	68
breng#nul#naar#buiten	64
houd#np#aan#kant	61
heb#np#in#voorbereiding	51
zet#np#op#voorsprong	36
ga#nul#naar#strand	35
kom#nul#op#podium	35
kom#nul#aan#kook	34
val#nul#buiten#boot	32
breng#np#te#weeg	31
kom#nul#op#rekening	31
schiet#nul#te#kort	31
zit#nul#in#film	29
kom#np#aan#kook	27
krijg#np#op#been	26
kom#nul#uit#dal	24
lijk#nul#van#plan	24
zet#np#op#agenda	24
krijg#nul#tot#taak	23
breng#np#om#zeep	21
haal#np#uit#kan	11
heb#np#in#aanbouw	11
houd#np#op#kier	11
krijg#np#op#orde	11
stuur#np#op#pad	11
verloop#nul#naar#wens	11
balanceer#nul#op#randje	10
doe#np#met#woord	10
ervaar#nul#aan#lijve	10
kom#np#boven#tafel	10
neem_op#np#in#wet	10
reken#nul#op#meerderheid	10
verhef#np#tot#kunst	10
zak#nul#naar#plaats	10

Table 16: False positives extracted using CO_2 classifier.

- the rather high proportion of annotation errors is likely to have a negative effect on the performance of the classifier. Errors in training data probably mislead the classifier in learning the concept we are interested in.
- many errors seem to include a predicative, locative or directional PP. These PPs exhibit a similar behavior than the actual PP argument in V (NP) PP MWES therefore, they are rather hard to classify.
- other errors seem to be metaphors and institutionalized phrases.

6 Conclusions

The MWE identification task is envisaged as a binary classification task. The classification is carried out by a well-studied machine learning algorithm known as decision trees (an implementation of the C4.5 algorithm). The best performing classifier reaches an accuracy of 83.4% in classifying both MWES and productive word combinations. Nonetheless, the classifier accuracy is expected to be higher due to a significant proportion of annotation errors in the training data. In addition, the accuracy may change slightly after the false positives are checked by human judges. Furthermore, the classifier does a good job in identifying MWES from *all* frequency ranges, including low-frequency expressions. This is a very positive feature. A less positive aspect is the precision and the recall achieved by the classifier in identifying MWES. Both scores are still rather low (0.66, 0.53, respectively). It may happen that once the annotation errors are eliminated, performance improves overall but this needs to be checked further.

An open question is whether identification of MWE can best be modelled as a binary classification of candidate expressions or as a probabilistic ranking of candidate expressions. The set of MWES existing in a language is very varied, with expressions exhibiting different behaviors at different levels of analysis. Although I try to capture this behavior with some quantitative techniques, adding yet another linguistic feature brings about little improvement in the performance. It is possible to engineer other attributes and investigate other linguistic properties, however, the gain seems to be very little. After a thorough evaluation of the performance of the decision trees classifier, big improvements on the performance seem unlikely. Perhaps, a binary decision (yes/no) is too difficult given that many expressions are located somewhere in between 'being a MWE' and 'could be a MWE'. This means that, perhaps a model that ranks candidate expressions according to the level of 'MWE-hood' is more appropriate. This observation has been made by Stevenson et al. [2004] regarding the lightness of LVC constructions; it probably applies to MWES in general. Seemingly, Gertjan van

Noord (p.c.) maintains that a binary classification is not an adequate model and a ranking of the candidates is desirable.

References

- Timothy Baldwin. Looking for prepositional verbs in corpus data. In *Proc. of the 2nd ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their use in computational linguistics formalisms and applications*, Colchester, UK, 2005.
- Hans Broekhuis. Het voorzetselvoorwerp. *Nederlandse Taalkunde*, 9(2):97–131, 2004.
- N. Calzolari, C.J. Fillmore, R. Grishman, N. Ide, A. Lenci, C. MacLeod, and A. Zampolli. Towards best practice for Multiword Expressions in computational lexicons. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1934–1940, Las Palmas, Canary Islands, 2002.
- Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74, 1993.
- A. Fazly and S. Stevenson. Automatically constructing a lexicon of verb phrase idiomatic combinations. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, Trento, Italy, 2006.
- Witten I. H. and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.
- Bart Hollebrandse. Dutch light verb constructions. Master’s thesis, Tilburg University, the Netherlands, 1993.
- Adam Kilgarriff and David Tugwell. Word sketch: Extraction & display of significant collocations for lexicography. In *Proceedings of the 39th ACL & 10th EACL -workshop ‘Collocation: Computational Extraction, Analysis and Exploitation’*, pages 32–38, Toulouse, 2001.
- W. Martin and I. Maks. *Referentie Bestand Nederlands. Documentatie*, April 2005.
- Paola Merlo and Matthias Leybold. Automatic distinction of arguments and modifiers: the case of prepositional phrases. In *Procs of the Fifth Computational Natural Language Learning Workshop (CoNLL-2001)*, pages 121–128, Toulouse. France, 2001.
- Tom Mitchell. *Machine learning*. McGraw Hill, 1997.

- P. Pecina and P. Schlesinger. Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 651–658, Sydney, Australia, 2006.
- Suzanne Stevenson, Afsaneh Fazly, and Ryan North. Statistical measures of the semi-productivity of light verb constructions. In Takaaki Tanaka, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors, *Second ACL Workshop on Multiword Expressions: Integrating Processing*, pages 1–8, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- Yee Fan Tan, Min-Yen Kan, and Hang Cui. Extending corpus-based identification of light verb constructions using a supervised learning framework. In *Proceedings of the EACL 2006 Workshop on Multi-word-expressions in a multilingual context*”, pages 49–56, Trento, Italy, 2006.
- Begoña Villada Moirón. *Data-driven Identification of fixed expressions and their modifiability*. PhD thesis, University of Groningen, 2005.
- Joachim Wermter and Udo Hahn. Collocation extraction based on modifiability statistics. In *Proceedings of Coling 2004*, Geneva, Switzerland, 2004. COLING.