What Is Beyond Collocations? Insights from Machine Learning Experiments

Leo Wanner*, Bernd Bohnet and Mark Giereth****
*ICREA and Pompeu Fabra University
Passeig de Circumval·laciô, 8
08003 Barcelona, Spain
**University of Stuttgart
Universitätsstr. 38
70569 Stuttgart, Germany

Abstract
Traditionally, collocations are treated in lexicography as idiosyncratic word combinations that must be learnt by heart by second language learners and which must thus be listed explicitly in collocation dictionaries. However, the learners' capacity to understand and to produce collocations they have never heard before indicates that collocations are not as opaque as often assumed. In our work on the extraction of collocations from corpora and their classification with respect to a fine-grained semantically-oriented typology, we experiment with several alternative machine learning techniques that exploit different characteristic features of collocations. These techniques can be viewed to model different strategies used by learners for the recognition of collocations. Their results can be thus expected to give us some evidence on how collocation dictionaries should be structured in order to provide best access to this important part of lexis.

1 Introduction

In lexicography, collocations are traditionally considered idiosyncratic word combinations which must be learned by heart by second language learners and which are, therefore, to be listed explicitly in collocation dictionaries. Consider, for instance, *give* [*a*] *lecture*, *take* [*a*] *walk*, *attend* [*a*] *conference*, etc.: in German, you 'hold' a lecture ([*eine*] *Vorlesung halten*) and in Russian you 'read' it (*Čitat' lekciju*); in German and French, you 'make' a walk ([*einen*] *Spaziergang machen*, *faire* [*une*] *promenade*), while in Spanish you 'give' it (*dar* [*un*] *paseo*); in German and Russian, you 'visit' a conference ([*eine*] *Konferenz besuchen*, *posetit' konferenciju*), while in Spanish, you 'assist' to it (*asistir* [*al*] *congreso*). The Oxford collocation dictionary, the BBI, and the Explanatory Combinatorial Dictionaries – to name just a few – are examples of such explicit collocation listings.

However, despite this obvious idiosyncrasy, second language learners often understand and produce collocations they have never heard before. How can this be explained? The answer to this question may well influence the design of the macrostructure of collocation dictionaries.

Obviously, collocations are semantically less opaque than we might assume at first instance. It what follows, we investigate how the semantic description serves the machine best in the context of automatic collocation understanding. By "understanding" we mean the identification of the semantics of collocations by automatically classifying them according to a fine-grained semantically-oriented collocation typology. From our findings, we expect to be able to draw conclusions concerning human processing of collocations. We explore the following three strategies:

(i) Classification by using prototypical samples for each type of collocations. When a new word bigram is to be classified, its semantic features are compared with the semantic features of the prototypical samples of each type in the typology. The bigram is assumed to be of the type the samples of which are most similar to the bigram.

(ii) Classification by using presumed characteristic semantic features of the elements of the samples for each type of collocations. When a new word bigram is to be classified, the semantic features of its elements are compared with the characteristic features of the samples collected for each type in the typology. The bigram is assumed to be of the type the characteristics of which are most similar to the bigram.

(iii) Classification by using a presumed characteristic correlation between the semantic features of the elements of the typical samples of each type of collocations. When a new word bigram is to be classified, the interdependency between the features of its elements is compared with the correlating features that are representative for each type in the typology. The bigram is assumed to be of the type the samples of which reflect the most similar correlation.

Each strategy has been implemented in terms of a distinct machine learning (ML-) technique; a series of experiments has been conducted with each of them. All experiments have been carried out with Spanish material. As collocation typology, we used the *lexical functions* (LFs) known from the Explanatory Combinatorial Lexicology (Mel'čuk, 1996). As the source of the semantic description of collocation elements, we used the Spanish part of the EuroWordNet lexical database (Vossen, 1998), henceforth SpEWN.

The remainder of the paper is structured as follows. In the next section, we briefly introduce the lexicological basics of our work. In Section 3, we present the ML-techniques used to implement the different strategies listed above. Section 4 contains a short overview of SpEWN. In Section 5 the experiments we carried out are outlined and their results are evaluated. Section 6, finally, concludes summarizing the most important findings of these experiments.

2 Lexicological and Formal Basics

In this section, we first introduce the notion of LFs, listing the LFs we refer to in the course of our presentation and present then the formal description of LF-instances as used in the sections on ML-experiments.

2.1 Lexical Functions

The following presentation of LFs is restricted to the absolute minimum necessary for the understanding of the presentation in the subsequent sections. Readers interested in a more profound introduction are referred to the numerous publications on LFs, and in particular, to (Mel'čuk, 1996).

In the context of collocations, only *syntagmatic* LFs are of relevance. A syntagmatic LF encodes a standard abstract lexico-semantic relation between two lexical units among which one of the units (the *base*) controls the lexical choice of the other unit (the *collocate*). "Standard" means that this relation is sufficiently common; "abstract" means that this relation is sufficiently generic to group all relations that possess the same semantic nucleus. We focus on standard abstract verb-noun relations. Typical examples of standard abstract relations between a noun and a verb are 'perform' (as between *give* and *presentation, make* and *suggestion, take* and *walk*, etc.) and its phasal counterparts 'start to perform' (as between *open* and *discussion, enter* [*into*] and *debate, get* and *headache*, etc.), 'continue to perform' (as between *retain* and *power, keep* and *influence, carry on* and *conversation*), and 'end to perform' (as between *lose* and *power, overcome* and *crisis, end* and *presentation*). In total, about twenty different verb-noun relations of this kind have been identified. For convenience, as names of LFs, Latin abbreviations are used. In our experiments, we used the following nine different LFs for which we give, in what follows, their semantic glosses and a number of examples:[1]

Oper1 'perform', 'experience', 'carry out', etc.; e.g.:
dar [*un*] *golpe* lit. 'give [a] blow', *presentar* [*una*] *demanda* lit. 'present [a] demand, *hacer* [*una*] *campaña* lit. 'do [a] campaign, *sentir* [*la*] *admiración* lit. 'feel [the] admiration', *tener* [*la*] *alegría* lit. 'have [the] joy'

ContOper1 'continue to perform', continue to experience', etc.; e.g.:
guardar [*el*] *entusiasmo* lit. 'keep [the] enthusiasm', *conservar* [*el*] *odio* lit. 'conserve [the] hatred, *pasar* [*la*] *vergüenza* lit. 'pass [the] shame'

Oper2 'undergo', 'be source of', etc.; e.g.:
someterse [*a un*] *análisis* lit. 'submit [oneself to an] analysis, *afrontar* [*el*] *desafío* lit. 'face [the] challenge', *hacer* [*un*] *examen* lit. 'do [an] examination', *tener* [*la*] *culpa* lit. 'have [the] blame'

Real1 'act accordingly to the situation', 'use as foreseen', etc.; e.g.:
ejercer [*la*] *autoridad* lit. 'exercise [the] authority', *utilizar* [*el*] *teléfono* lit. 'use [the] telephone', *hablar* [*una*] *lengua* lit. 'speak [a] language, *cumplir* [*la*] *promesa* lit. 'fulfil [the] promise'

Real2 'react accordingly to the situation'; e.g.:
responder [*a la*] *objección* lit. 'respond [to the] objection', *satisfacer* [*el*] *requisito* lit. 'satisfy [the] requirement', *atender* [*la*] *solicitud* lit. 'attend [the] petition', *rendirse* [*a la*] *persuasion* lit. 'render (oneself) [to the] conviction'

CausFunc0 'cause the existence of the situation, state, etc.'; e.g.:
dar alarma lit. 'give alarm', *celebrar elecciones* lit. 'celebrate elections', *publicar* [*una*] *revista* 'publish [a] journal', *provocar* [*una*] *crisis* lit. 'provoke [a] crisis'

[1] The subscripts the LF-names specify the projection of the semantic structure of the collocations denoted by an LF onto their syntactic structure. In our experiments, we interpret complete LF-names as collocation class labels. Therefore, we can ignore the semantics of the subscripts and consider them simply as part of LF-names. Recall that we are working with Spanish material. Therefore, we provide here Spanish examples.

FinFunc0 'the situation ceases to exist'; e.g.:
[la] aprensión se disipa lit. '[the] aprehension evaporates',

Caus2Func1 'cause (by the object) to be experienced / carried out / performed'
dar [una] sorpresa lit. give [a] surprise, *provocar [la] indignación* lit. 'provoke [the] indignation', *despertar [el] odio* lit. 'awake [the] hatred'

IncepFunc1 'begin to perform / to experience / to carry out'; e.g.:
[la] desesperación entra [en N] lit. '[the] despair enters [in N]', *[el] odio se apodera [de* N] lit. '[the] hatred gets hold [of N]', *[la] ira invade* [N] lit. '[the] rage invades [N]'

2.2 Basic assumptions and notations

Our work is grounded in the assumption that collocations may receive a componential description. They are what Baldwin et al. (2003) call "simple decomposable multiword expressions". For our purposes, we use a semantic component description of collocation elements. That is, in a collocation B⊕C (with B being the meaning of the base *B*, C the meaning of the collocate *C* and B⊕C the meaning of the collocation *B⊕C* as a multiword unit), B is assumed to be given by the set of components $\{b1,b2,\ldots,bNb\}$ and C by the set of components $\{c1,c2,\ldots,cNc\}$ ('*Nb*' stands for the number of components in the base description and '*Nc*' for the number of components in the collocate description). The componential description of lexical meanings is expected to be available from an external lexical resource. Any sufficiently comprehensive lexico-semantic resource suitable for NLP can be used; as already mentioned, we use the Spanish part of EuroWordNet, SpEWN. The componential meaning descriptions facilitate the use of machine learning techniques for the implementation of the three above collocation classification strategies in that they allow for the derivation of an explicit and verifiable correlation either between subsets or complete sets of base meaning components and subsets / sets of collocate meaning components characteristic of a given LF.

To learn a correlation between the semantics of a base and the collocates this base co-occurs with, we start from a training set of manually compiled disambiguated instances for each of the *n* LFs used for classification. That is, if in an LF-instance *B⊕C* contained in a training set, *B* and/or *C* are polysemous, only the description of one sense of *B* (the one which comes to bear in *B⊕C*) and the description of one sense of *C* are taken.

Before we enter into the presentation of the machine learning techniques in the next section, let us introduce the notations and abbreviations used henceforth:

- a base lexeme is referred to as *B* and a collocate lexeme as *C*; accordingly, the meaning description of *B* is defined as B = $\{b1,b2,\ldots,bNb\}$ and the meaning of *C* as C = $\{c1,c2,\ldots,cNc\}$;
- a collocation instance in a training set for a given LF is referred to as (*B*,*C*) and its meaning as (B,C) or B⊕C;
- given a training set of instances for each LF **L1,L2,…,Ln** in the typology, B stands for the meaning component collection over the base sets of the instances from the training sets of all LFs and C for the meaning component collection over the collocate sets of the instances from the training set of all LFs;
- a candidate noun-verb bigram that is to be classified (recall that we concentrate on noun-

verb collocations) is referred to as (N,V), the meaning description of the noun N as N = $\{n1,\ldots,nNN\}$, and the meaning description of the verb V as V = $\{v1,\ldots,vNV\}$.

3 Implementing Collocation Classification Strategies by ML-Techniques

Let us now introduce the three ML-techniques we use to model the different collocation recognition (= classification) strategies listed in Section 1.

3.1 Classification by Using Prototypical Collocation Samples

For the realization of the classification of collocations by using prototypical samples for each LF (i.e., for each type of collocations), the so-called "nearest neighbour" (NN) technique is suitable. This technique compares the candidate bigrams with the training instances, choosing for each bigram one or several instances that are most similar ("nearest") to it. The bigram is assumed to belong to the same class (be of the same type) as its nearest instance. If several nearest instances are being selected, a voting procedure may be implemented: the candidate bigram is assigned to the class to which the majority of the nearest instances belong.

Unlike the other ML-techniques, NN-classification does not include, strictly speaking, a learning stage. Rather, it can be thought of as consisting of a training material representation stage and a classification stage.

The representation of the training material for NN-classification can in abstract terms be described as a pair of vector space models (Salton, 1980) – a base vector space and a collocate vector space: assume a training set of instances for each LF **L1**, **L2**, …, **Ln** in the typology; the corresponding B and C naturally map onto multidimensional vector spaces *VB* (the base description space) and *VC* (the collocate description space). Each component $b \in$ B and each component $c \in$ C provides a distinct dimension in *VB* and *VC*, respectively. Each training instance I is thus represented by a pair of vectors $(\rightarrow vbI, \rightarrow vcI) \in (VB, VC)$. In the simplest realization of the model, $\rightarrow vbI$ and $\rightarrow vcI$ will contain a '1' for dimensions (= components) available in I and a '0' for dimensions that are not available in I. Obviously, realizations with a weighting schema are possible to take into account the varying importance of dimensions for the description of a collocation. We use a binary weighting schema.

Before applying this representation in the classification stage, those samples may be removed from (B, C) that are "unreliable". As unreliable, we consider a sample if it is nearest to an instance of a different LF than it is itself. To determine which instance is nearest, we use equation (1) from the classification stage; see below.

Given a candidate word bigram $K := (N, V)$ that is to be classified according to the LF-typology, the classification stage consists of (i) decomposition of the meaning of N and V as (N,V), and (b) mapping of (N,V) onto (*VB*, *VC*). The LF-label of the instance I whose vector pair $(\rightarrow vbI, \rightarrow vcI)$ is nearest to the vector pair $(\rightarrow vnK, \rightarrow vvK)$ of K is assigned to the candidate.

To determine the similarity between $(\rightarrow vbI, \rightarrow vcI)$ and $(\rightarrow vnK, \rightarrow vvK)$, the cosine or any other suitable metric can be used. In our experiments, we used the following set-based metric:

(1) $sim(I,K) = b\, fb\, /\, fbmax|N| + g\, fc\, /\, fcmax\, |V|$

with *fb* as the number of dimensions shared by →*vbI* and →*vnK*; *fbmax* as the maximal number of dimensions shared by →*vnK* and a base vector of any instance in the training set for the LF of which *I* is an instance, *fc* as the number of dimensions shared by →*vcI* and →*vvK* and *fcmax* the maximal number of dimensions shared by →*vvK* and a collocate vector of any instance in the training set for the LF of which *I* is an instance. |*N*| stands for the number of components in the description of the noun of *K* and |*V*| for the number of components in the description of the verb. *b* and *g* are constants that can be used to tune the importance of the base and collocate, respectively, for the classification task. In our experiments, we used *b* = 1 and *g* = 1.5; that is, we assigned higher importance to the collocate meaning than to the base meaning. If *fcmax* = 0 (which means that →*vcI* and →*vvK* do not share any dimension), the second summand in Equation (1) becomes invalid and the candidate bigram is rejected as a collocation of the type **L** of *I*. The candidate is also rejected if *sim*(*I*,*K*) is smaller than a given threshold for all instances of **L** in the training set.

3.2 Classification by Using Characteristic Semantic Features of Collocation Elements

A series of ML-techniques is available that use isolated characteristic features of collocation elements, i.e., that do not take the interdependency between the features (e.g., between a prominent base feature and a prominent collocate feature or between two prominent collocate features) into account. We have taken the popular Naïve Bayes classification technique.

The central part of any Bayes classification technique is the so-called *Bayesian network*. A general Bayesian network can be viewed as a labelled directed acyclic graph that encodes a joint probability distribution over a set of random variables V = {*X1*,*X2*,...,*Xn*}. When used for classification, usually a class variable (here the LF-variable) and a number of attribute variables (here, semantic component labels) are introduced. The value of the variables may be again either '1' or '0'. The names of the variables function as labels of the nodes of the graph; the co-occurrence dependency between variables is represented by arcs connecting the nodes they label.

The Naïve Bayesian network is the simplest realization of a Bayesian network. It assumes that the attribute variables depend only on the class variable; attribute variables are mutually conditionally independent. The network is thus restricted to a tree of depth 1, with the LF-variable as the root node, component variables as the attribute leaf nodes, and edges defined from the class node to attribute nodes. For each instantiation of the LF-variable, i.e., for each LF in the typology, the edge between the LF-variable and any attribute node is labelled by the probability that the corresponding component occurs in the description of the samples of the LF in question. The probability is calculated based on the component distribution within the samples in the training set for LF. For the classification of a given noun-verb bigram (*N*,*V*), the joint probability over all components that occur in the descriptions of *N* and V, i.e., N and V, is computed for each LF. The LF with the highest probability is selected as label for (*N*,*V*).

For readers interested in technical details, some more formal information might be of relevance. Thus, to compute the probability of each potential LF-label L, we apply the *Bayes rule*. The label with the highest posterior probability is then predicted to be the LF-label for (*N*,*V*), i.e.:

(2) CLF = argmaxLFj P(N "* V)|LFj) = ΠcoN ∪ V *P*(co|LFj)

where CLF is the most probable class variable value, and where LFj ranges over all LFs in the typology. Given that the attributes are considered independent, P(co|LFj) for any component co can be estimated adopting the m-estimate of probability (Mitchell, 1997, pp. 179,182):

(3) $P(co|LFj) = (nkco + 1) / (nco + |B \cup C|)$

where nco is the total number of components in the descriptions of all training examples whose class variable value is LFj and nkco is the number of times the component co is found among these nco components. |B ∪ C| stands for the total number of distinct component in the training set descriptions.

3.3 Classification by Using the Correlation between Features of Collocation Elements

The Naïve Bayesian Network attempts to grasp the characteristic features of the collocation elements. Intuitively, however, it is the correlation between the semantic features of the collocation elements that is important. Mel'čuk and Wanner (1996) demonstrated that such a correlation exists and that this correlation can be used, for instance, for the definition of an inheritance-oriented macro structure in collocation dictionaries.

An ML-technique that allows us to model this correlation is the *Tree-Augmented Network* (TAN) Classification technique (Friedman et al, 1997). TAN is an extension of the NB-classification technique. The structure of a TAN is based on the structure of the Naïve Bayesian network, i.e., it also requires that the class variable node be parent of every attribute node. But to capture the correlations between the components, additional edges between attribute nodes are introduced, which are labelled by the component co-occurrence probabilities within the descriptions of the samples of an LF.

To take into account that the correlation between components depends on the LF in question (i.e., the value of the class variable), we construct for every instantiation of the LF-variable a TAN. This "multinet" extension of the original TAN-classifier is also along the lines of the proposal in (Friedman et al, 1997).

In order not to make the presentation more technical than necessary, we dispense with the presentation of the algorithm for the construction of TANs; the interested reader is asked to consult, e.g., (Cheng and Greiner, 2001) or any other of the numerous publications on the topic. Given the structure of a multinet, the formula used to classify a candidate bigram (N,V), the class variable value *LFk* with the most optimal network is chosen:

(4) *CLF = argmaxLFk P(LFk) =* P*co1, co2*∈**N» V** *P(co|LFj) IP(co1,co2| LFk)*

with *IP(co1,co2| LFk)* as the conditional mutual information between two meaning components *co1* and *co2*, given *LFk.*

4 Spanish EuroWordNet

1As already mentioned, for the componential description of the LF-instances in the training sets as well as for the description of the candidate bigrams, we use the Spanish part of the

EuroWordNet (EWN), henceforth SpEWN. More precisely, we use the *hyperonymy hierarchies* of lexical items provided by SpEWN. EWN is a multilingual lexical database which comprises lexico-semantic information organized following the relational paradigm (Vossen, 1998). The current version of the SpEWN has to a major part been derived automatically from the English WordNet developed at Princeton University (Fellbaum, 1998). In contrast to the original Princeton WordNet, where the hyperonymy hierarchy of a lexical item is purely lexical (i.e. contains only hyperonyms), in SpEWN (as in most WNs in the EWN), the hyperonym hierarchy of each lexical item consists of:

- its hyperonyms and synonyms (i.e., words that combine with the lexical item in question to form a (*synset*)
- its own *Base Concepts* (BCs) and the BCs of its hyperonyms
- the *Top Concepts* (TCs) of its BCs and the TCs of its hyperonyms

Figure 1 shows, for illustration, the hyperonym hierarchies (including synonyms, BCs and TCs) of PRESENTAR1 'present' and RECLAMACIÓN3 'declaration' from the collocation *presentar* [*una*] *declaración* lit. 'present [a] reclamation' ('lodge [a] reclamation').

```
((7. communication RECLAMACIÓN3
    6. communication INSTANCIA2 PETICIÓN1 PEDIDO1
        5. communication Communication | Mental | Usage CONTENIDO3 MENSAJE2
          4. Tops 3rdOrderEntity | Communication | Mental | Purpose | Social COMUNICACIÓN1
              3. Tops Relation | Social RELACIÓN-SOCIAL1
                 2. Tops Relation RELACIÓN1
                    1. Tops ABSTRACCIÓN1)
(6. communication PRESENTAR3
   5. communication SOMETER2
      4. communication Agentive | BoundedEvent | Communication | Purpose PEDIR1
         3. communication Agentive | Communication | UnboundedEvent COMUNICAR2
            2. social Agentive | Dynamic | Social INTERACTUAR1
               1. social Agentive | Dynamic ACTUAR4 LLEVAR-A-CABO2 HACER15))
```

Figure 1. Hyperonym hierarchies for PRESENTAR3 and RECLAMACIÓN3 in the collocation *presentar* [*una/la*] reclamación (lexical items are written in small capitals, BCs and TCs are in sans serif, and the TCs start with a capital; individual TCs are separated by the '|' sign)

BCs are general semantic labels that subsume a sufficiently large number of synsets. Examples of such labels are: change, feeling, motion, and possession. Thus, DECLARACIÓN3 'declaration' is specified as communication, MIEDO1 'fear' as feeling, PRESTAR3 'lend' as possession, and so on.

Unlike *unique beginners* in the original WN, BCs are mostly not "primitive semantic components" (Miller, 1998); rather, they can be considered labels of semantic fields. The set of BCs used across different WNs in the EuroWN consists of 1310 different tokens. The language-specific synsets of these tokens constitute the cores of the individual WNs in EuroWN.

Each BC is described in terms of TCs – language-independent features such as Agentive, Dynamic, Existence, Mental, Location, Social, etc. (in total, 63 different TCs are distinguished). For instance, the BC change is described by the TCs Dynamic, Location, and Existence.

5 Experiments and their Evaluation

We conducted first two experiments with different training and test material. In the first experiment, we trained on and classified verb-noun bigrams whose nouns all belong to the same semantic field, namely to the field of emotion nouns. In the second experiment, we trained on and classified verb-noun bigrams with no consideration of field constraints. A separate experiment on mono-field material is of value because the semantics of the nouns that belong to the same semantic field are a priori homogeneous at a certain level of abstraction. The lexical-semantic description of the instances of the same LF can thus be assumed to be similar. We may also hypothesize that for second language speakers it is easier to handle new collocations if they belong to the same semantic field as those they already know. We have chosen emotion nouns because they are rich in collocations and because for emotion nouns lists of LF-instances are already available for Spanish (see below).

Intuitively, the more collocations we know as language learners the better we can correctly interpret new unknown ones. In accordance with this assumption, the classification experiments in Section 5.1 have been carried out with 95% of the samples available for each LF as training material and 5% as test material. However, this assumption presupposes that the learning material is balanced; i.e., that we progressively learn instances of all LFs. If this is not assured, we might become biased towards one of the LFs. In order to get some experience on this aspect of collocation learning, we also experimented with different training set ratios; cf. Section 5.3.

Finally, one must be aware that each collocation recognition strategy from Section 1 can be implemented by a number of different machine learning techniques. Each of these techniques may have its own peculiarities and lead thus to different results. For illustration, we show the results achieved for classification of both emotion noun and field independent bigrams by a second technique that uses isolated characteristic features of collocation elements – a decision tree classification technique based on the ID3-algorithm (Quinlan, 1986); cf. Section 5.4

5.1 Classification Experiments

For Experiment 1, we used the following five of the nine LFs listed in Section 2: Oper1, ContOper1, Caus2Func1, IncepFunc1 and FinFunc0; for Experiment 2, we used CausFunc0, Oper1, Oper2, Real1 and Real2. For glosses and examples for each of these LFs, see Section 2. Tables 1 and 2 give information on the number of the instances used for each LF in the experiments. For Experiment 1, a collection of Spanish collocations from (Alonso Ramos, 2003; Sanromán, 2003) that are already classified in terms of LFs has been used. For Experiment 2, the data have been collected by interviewing native speakers of Spanish and by consulting dictionaries.

$\text{Caus}_2\text{Func}_1$	ContOper_1	FinFunc_0	Oper_1	IncepFunc_1
71	14	40	37	23

Table 1. Distribution of LF-instances in Experiment 1

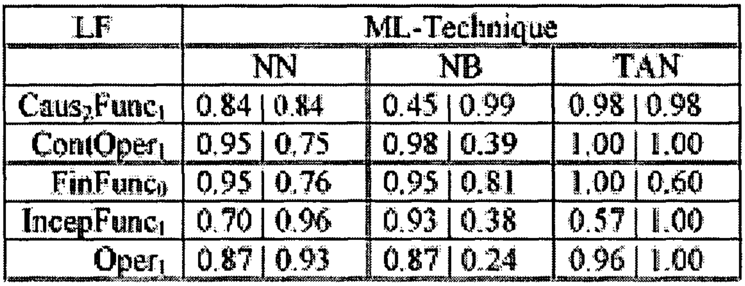
CausFunc_0	Oper_1	Oper_2	Real_1	Real_2
53	87	48	52	53

Table 2. Distribution of LF-instances in Experiment 2

All experiments have been carried out with non-disambiguated test material.[2] Given that in SpEWN an element of any test bigram usually has more than one sense, the cross-product of all possible readings of each test bigram must be built. That is, if we assume that for a given bigram (*N*,*V*), the noun *N* encounters *sN* senses and the verb *V sV* senses, we build $\{Se1N, Se2N, \ldots, SesNN\} \times \{Se1V, Se2V, \ldots, SesVV\}$, where *SeiN* ($1 \leq i \leq sN$) is one of the nominal senses and *SejV* ($1 \leq j \leq sV$) one of the verbal senses. To classify a given candidate bigram, (*SeiN*, *SejV*)s of this word bigram are examined as prescribed by the ML-techniques in use. Obviously, only one of the (*SeiN*, *SejV*)s may qualify the word bigram as an instance of a specific LF. However, as is well-known, the distinction of word senses in SpEWN is biased towards English, which means that sense distinctions are made for a Spanish word if the corresponding readings are available for the English material – even if they are not available in Spanish; cf. (Wanner et al. 2004) for examples. As a result, Spanish words are often assigned several incorrect senses. This has negative consequences for the quality of the classification procedure. To minimize these consequences we use for all ML-techniques the so-called "voting" strategy: instead of choosing ONE sense bigram as evidence that the word bigram is instances of the LF L, each sense bigram of the given word bigram "votes" for an LF; the word bigram is assigned the LF-label with most votes.

To eliminate a distortion of the experiment outcomes by the selection of the training samples, experiments are run in 200 to 500 iterations. The quality figures cited below reflect the average performance over all iterations.

[2] Recall, however, that we train on manually disambiguated LF-instances.

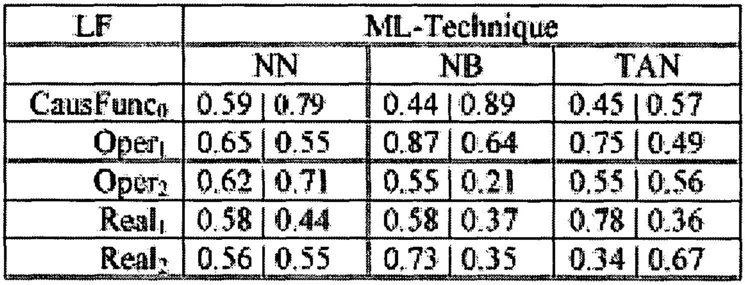


LF	ML-Technique		
	NN	NB	TAN
Caus$_2$Func$_1$	0.84 | 0.84	0.45 | 0.99	0.98 | 0.98
ContOper$_1$	0.95 | 0.75	0.98 | 0.39	1.00 | 1.00
FinFunc$_0$	0.95 | 0.76	0.95 | 0.81	1.00 | 0.60
IncepFunc$_1$	0.70 | 0.96	0.93 | 0.38	0.57 | 1.00
Oper$_1$	0.87 | 0.93	0.87 | 0.24	0.96 | 1.00

Table 3. The quality figures (as $p|r$) of emotion noun bigrams by the different ML-techniques

Tables 3 and 4 show the performance of the three ML-techniques. 'p|r' stands for 'precision|recall'. As usual, we define precision as $p = |LFci| / |LFpe|$ and recall as $r = |LFci| / |LFi|$, where $|LFci|$ is the number of test set elements correctly classified as the LF i, $|LFpe|$ the total number of test set elements classified as the LF i, and $|LFi|$ the total number of test set elements available for the LF i.

The tables reveal that both the absolute and comparative performance of the techniques varies from LF to LF. No technique can be identified as the best at the first glance. Let us evaluate thus in the next subsection the performance of each technique in more detail with respect to its precision and recall.



LF	ML-Technique		
	NN	NB	TAN
CausFunc$_0$	0.59 | 0.79	0.44 | 0.89	0.45 | 0.57
Oper$_1$	0.65 | 0.55	0.87 | 0.64	0.75 | 0.49
Oper$_2$	0.62 | 0.71	0.55 | 0.21	0.55 | 0.56
Real$_1$	0.58 | 0.44	0.58 | 0.37	0.78 | 0.36
Real$_2$	0.56 | 0.55	0.73 | 0.35	0.34 | 0.67

Table 4. The quality figures (as $p|r$) of field-indipendent bigrams by the different ML-techniques

5.2 Classification Experiment Evaluation

The performance of all techniques varies considerably between Experiment 1 and Experiment 2. While for emotion bigrams, especially TAN achieves the optimal quality figures for several LFs, for field-independent bigrams no techniques reaches a 100% quality.

5.2.1 Emotion bigram classification evaluation

Table 3 illustrates that in the case of emotion collocation classification, the precision of TAN is considerably better than that of NN and NB for Caus2Func1 and Oper1; it is still better (although less so) for ContOper1 and FinFunc0, and it is worse than NN and NB for IncepFunc1. NB performs better than NN for ContOper1 and IncepFunc1, while NN is considerably better than NB for Caus2Func1. For FinFunc0 and Oper1, NN and NB perform the same. To understand why, we must assess the semantic characteristics of samples used in the training and test material; especially the analysis of the collocates may be instructive. We presuppose that the compilation of the training and test material has not been biased.

We have analyzed the semantic composition of the collocate elements available for each LF with respect to *similarity* and *dispersion*. The measure of similarity captures how similar the semantic descriptions of the samples for a given LF among each other are. The measure of dispersion captures to what extent semantic features of the samples for a given LF are encountered in the descriptions of samples for other LFs.

The performance of all techniques suffers from low similarity and high dispersion. However, NB is especially predisposed since it attempts to identify a number of isolated semantic features that are characteristic of each LF. Therefore, NB's precision is only high in the case of those LFs whose samples show a low dispersion and high similarity. In the case of Caus2Func1, the dispersion is high; therefore, NB performs poorly. NN and TAN are more stable than NB. TAN suffers from the dispersion at a higher level, namely the dispersion of feature co-occurrence. Thus, it performs worst for IncepFunc1 because samples of this LF are dominated by the interdependency of features that also often co-occur in the samples of other LFs. NB is not vulnerable to feature interdependency dispersion because it considers the probability of the occurrence of isolated features. As TAN's, NN's performance is also the lowest for IncepFunc1. However, it is not as low since the matching of the features of a candidate bigram with ALL features of prototypical samples helps reduce the bias towards high frequency co-occurrence.

The ML-techniques vary with respect to recall even more than they do with respect to precision. Also, the tendency of NB's performance to vary across different LFs is stronger; cf. 0.99 for Caus2Func1 and 0.24 for Oper1. NB's recognition of ContOper1-, IncepFunc1, and Oper1-instances is poor. This is because only a small share of isolated features is really specific to the instances of one LF. All others are shared by some instances of other LFs. If there are more than a few of such instances and there is more training material for the other LFs, NB becomes biased towards the other LFs. TAN achieves a maximal recall (1.00) for three LFs out of five. However, for FinFunc0, "only" 0.60 are achieved. This is because of the low discriminatory potential of the characteristic feature co-occurrences within the samples for this LF. NN's recall is rather high for all five LFs. This implies that for each LF instances are available which are prototypical enough to serve as reference collocations.

5.2.2 Field Independent bigram classification

Due to semantically more heterogeneous instances of the individual LFs it is not surprising that the performance of the techniques on field independent bigrams is worse than on emotion bigrams. The highest precision is achieved by NB with 0.87 for Oper1. NB is also better than NN and TAN for Real2, while TAN performs better for Real1, and NN for CausFunc0. The low performance of TAN for CausFunc0 and Real2 is again due to the low discriminatory potential of the most salient feature co-occurrences (i.e., high co-occurrence dispersion) within the samples for these two LFs. Also, TAN is biased towards classifying candidate bigrams as instances of CausFunc0 due to the numerical dominance of CausFunc0 samples in our experiment setting. This bias is also one of the reasons for the even poorer performance of NB towards CausFunc0. NN is again the most stable technique.

Despite the more heterogeneous instances available for the individual LFs, and thus less pronounced prototypical samples, the recall with NN is substantial. Only for Real1, it goes down to 0.44. This is because of the extreme diversity of the instances of Real1. Compare, e.g., *ejercer autoridad* lit. 'exercise authority' vs. *lanzar* [*un*] *misil* lit. 'lance [a] missile' vs. *pilotar* [*un*] *avión* lit. 'pilot [a] plane'. TAN also achieves its lowest recall for Real1 (even considerably lower than NN). NB's recall is highest for CausFunc0. This can be explained by the prominence of some of the features of many CausFunc0-instances in our training and test material. With respect to Oper2, NB reaches only a very low recall (namely, 0.21). Such a low recall is partially due to the more scarce presence of Oper2-samples in our material.

5.3 Training Set Ratio Experiments

As mentioned above, we tested the variation of the performance of the ML-techniques in relation to the proportion of the sizes of the training set and test set. In both experiments (emotion field oriented and field-independent experiments), for each LF, x% of the available LF-samples have been used as training material; the remaining 100 – x% of the samples of all five LFs used in the experiment served as test material. Experiments have been performed with x = 5%, 10%, 25%, 50%, 75% and 95%. To eliminate a distortion of the experiment outcomes by the selection of the training samples, for each ratio, experiments have again been run in 200 to 500 iterations. Figures 2-6 show the evolution of the precision (*p*) over the training set ratio for emotion bigram classification. Due to the lack of space, we do not discuss in detail the evolution of recall (*r*) and the evolution of *p* and *r* in the case of field-independent classification.

p varies for each LF and again depends on the ML-technique. In the case of NN-classification, for all LFs, except for ContOper1, the ratio of 10% provides the highest precision: 0.95 for Caus2Func1, 1.00 for FinFunc0, 0.73 for IncepFunc1, and 1.00 for Oper1.[3] This means that when 10% of the material available for the LF *i* is taken for training, the share of training instances for the LF *i*′ which are semantically similar to candidate bigrams for *i*, is the smallest. For ContOper1, the ratio of 95% leads to a significantly better *p* than 10%, which is the second best (0.95 compared to 0.93).

In case of NB, the precision for ContOper1, FinFunc0, IncepFunc1 and Oper1 increases significantly with the increasing training set size ratio.[4] The precision for Caus2Func1 does not improve over the increasing training set size ratio. The classification trace shows that this

[3] We use an equal weighting of p and r to calculate the *f*-score: $f = 2pr / (p + r)$.

[4] The decrease of *p* for the 10% ratio in the case of ContOper1 is due to our unfortunate experiment setup: given that we use only a few ContOper1-instances, the 5% and the 10% ratios for ContOper1 are equally small, while for the other LFs, the number of training instances is higher for the 10% ratio. This disproportion leads to the bias of the NB-network in particular towards CausFunc1 and FinFunc0.

is first of all because (*N*,*V*)s with a specific *V* (e.g., *sembrar* and *dar*) have notoriously been classified as FinFunc0. The componential descriptions of the majority of the senses of this *V* are too similar to the descriptions of FinFunc0 training set instances.

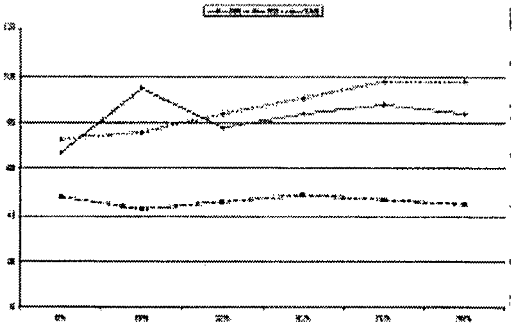


Figure 2. *p* of emotion bigram classification with Caus2Func1

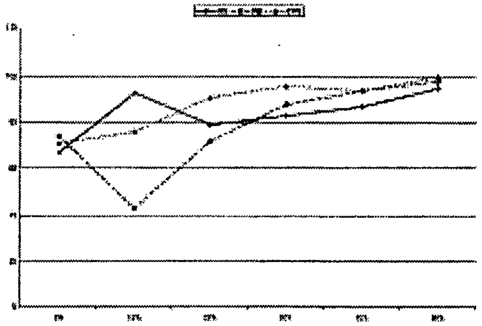


Figure 3. *p* of emotion bigram classification with respect to ContOper1

In the case of TAN-classification, Caus2Func1-, ContOper1-, FinFunc0, and Oper1-figures improve in general over the training set size ratios. For IncepFunc1, the highest precision is achieved with the training set ratio of 50%. With the 95% ratio, the network's performance decreases slightly to 57%. This means that with a ratio of 95%, instances come to bear whose characteristic features are met in instances of other LFs as well.

In general, it can be stated that while NB and TAN benefit from larger training sets, the NN-technique is rapidly saturated: as long as there are only a few prototypical instances for each LF, the correct prediction that a candidate bigram is sufficiently similar to one of these instances is easier than when there are more (and thus also more heterogeneous) prototypical instances. Obviously, this is only valid (as we will see immediately below) if the few prototypical instances are indeed representative for an LF in the given field.

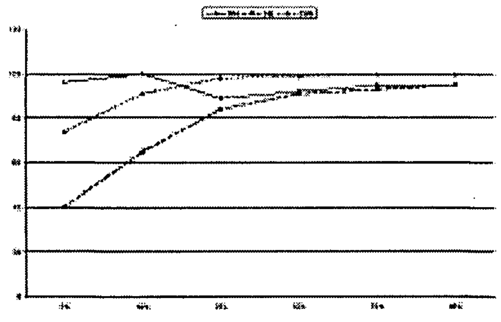


Figure 4. *p* of emotion bigram classification with respect to FuncFunc0

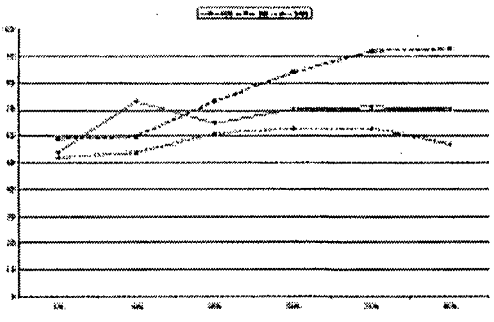


Figure 5. *p* of emotion bigram classification with respect to IncepFunc1

To contrast the evolution of the performance of the ML-techniques on emotion bigrams with their performance on field-independent bigrams, Figure 7 shows the precision of our techniques with respect to the Oper1-classification when applied to field-independent bigrams.

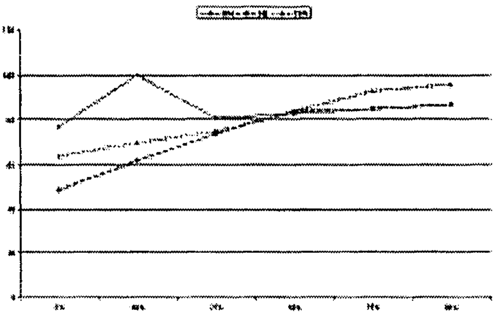


Figure 6. *p* of emotion bigram classification with respect to Oper1

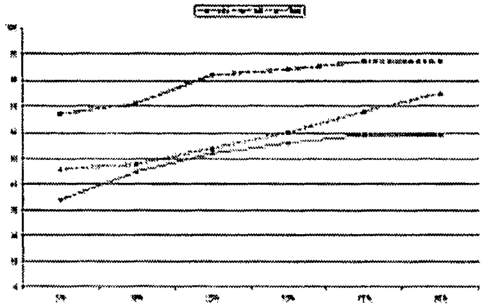


Figure 7. *p* of field-independent bigram classification with respect to Oper1

Unlike with respect to emotion bigrams, the precision of NN in Figure 7 steadily increases with the increasing training set ratio. This indicates that the prototypical collocations added to the training set of an LF (here, Oper1) at each stage of the training set ratio augmentation tend to cover new profiles of instances of this LF.

5.4 ID3 Classification Experiment

Let us now discuss how characteristic features of collocation elements can be used by another ML-technique, namely the ID3-based classifier, and compare the performance of ID3 to the performance of NB.

The ID3-algorithm constructs an optimal decision tree from the training instances available for each class (in our case, LF). The leaf vertices of the tree are class labels (= LF-names); the root and intermediary vertices are attribute (= meaning component) labels. The construction of the tree is recursive. During each recursion, the attribute with the minimal average entropy from the remaining list of not yet inserted attributes in the considered path from the root is chosen.[5]

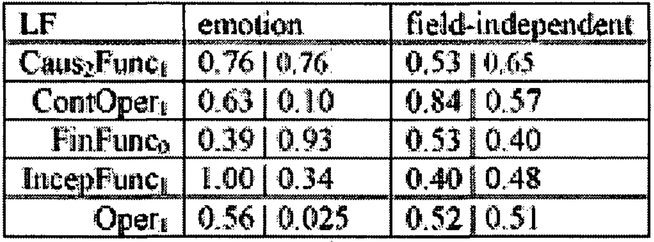
The performance of the decision tree constructed by ID3 decisively depends on the optimal determination of the attributes used as vertices of the tree. Since in our experiments each component *c* in B» C is a binary attribute, the resulting tree is a binary tree with '1' and '0' as labels of outgoing edges of an attribute vertice. As already discussed above, for an instance *I*

[5] We refrain from introducing here the algorithm and the average entropy formula used for attribute selection; the interested reader is asked to consult (Quinlan, 1986) or any introductory book on Machine Learning.

in a training set of the LF L, *c* has the value '1' if it is available in the description of *I*, and '0' if it is not available. When traversing the tree during classification, the '1'-path is taken if the description N» V contains the attribute in question and the '0'-path otherwise.

Table 5 shows the precision and recall achieved by the ID3-classifier with the 95% training set ratio. For emotion noun bigrams, ID3's precision is higher for Caus2Func1 and slightly also for IncepFunc1 than it was for NB (0.76 compared to 0.45 and 1.00 compared to 0.93). For the other three LFs, ID3's precision performance is worse. Its recall is in general lower, only for FinFunc0, it performs somewhat better. Note also the analogy in the distribution of the recall over the different LFs. For field-independent bigrams, ID3's precision is in general somewhat lower than NB's – except for CausFunc0 for which it reaches 0.53 (compared to 0.44). In contrast, its recall is higher for three of the LFs – FinFunc0, IncepFunc1 and Oper1.

To explain the deviating tendencies as well as the absolute differences between the performance of NB and ID3, a further thorough evaluation is needed.



LF	emotion		field-independent	
$\text{Caus}_2\text{Func}_1$	0.76	0.76	0.53	0.65
ContOper_1	0.63	0.10	0.84	0.57
FinFunc_0	0.39	0.93	0.53	0.40
IncepFunc_1	1.00	0.34	0.40	0.48
Oper_1	0.56	0.025	0.52	0.51

Table 5. Performance of ID3 (in terms of $p|r$) on emotion and field independent bigrams

6 Conclusions

Our experiments demonstrated that ML-techniques can be used to automatically classify collocations according to such a fine-grained semantic typology of collocations as lexical functions. Furthermore, our experiments have shown that the performance of the different techniques may vary considerably. Especially the performance of NB, but also that of TAN, seems very prone to the semantic profiles of the training and test instances of the LFs. NB is precise in the collocation recognition if many instances of a given LF reveal the same characteristic semantic features. For instance, for emotion bigrams, the characteristic features of ContOper1-collocations are, among others, 'Relation' and 'mantener' 'keep'. TAN is able to reach the highest quality figures of all techniques, but the variation of its performance makes it unreliable. NN is the most stable technique that we investigated, although its performance is in certain constellations somewhat lower than the performance of NB and/or TAN. This makes it at this stage of investigation the favorite ML-technique for collocation dictionary compilation. However, a deeper contrastive evaluation of the figures obtained by each technique is still necessary. Also, further experiments with additional collocation material must be carried out.

With respect to the structure and content of collocation dictionaries, our experiments demonstrated that it is useful to include information on prominent semantic features and their interdependency common to the instances of an LF. If an LF possesses outstanding prototypical samples, they should be also included. This would be helpful for the learner with respect to both decoding and encoding.

References

Alonso Ramos, M. (2003), 'Hacia un *Diccionario de colocaciones del español* y su codificaci6n', in Martí, M. A. et al. (eds.), *Lexicografía computacional y semdntica*, Barcelona: Universitat de Barcelona, pp. 11-34.

Baldwin, T., Bannard, C., Tanaka T., Widdows D. (2003), 'An empirical model of multiword expression decomposability', in Proceedings of the Workshop on Multiword Expressions at ACL 03.

Cheng, J., Greiner, R. (2001), 'Learning Bayesian Belief Network Classifiers: Algorithms and System', *Proceedings of the 14th Canadian conference on artificial intelligence.*

Fellbaum, Ch. (ed.) (1998), *WordNet. An Electronic Lexical Database.* Cambridge, MA: The MIT Press.

Friedman, N., Geiger D., Goldszmidt M. (1997), 'Bayesian network classifiers', *Machine Learning.* Vol. 29.2-3:131-163.

Mel'čuk, I.A. (1996), 'Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon', in Wanner, L. (ed.) *Lexical Functions in Lexicography and Natural Language Processing*, Amsterdam, Benjamins Academic Publishers, pp. 37-102

Mel'čuk, I.A., Wanner, L. (1996), 'Lexical Functions and Lexical Inheritance for Emotion Lexemes in German', in Wanner, L. (ed.) *Lexical Functions in Lexicography and Natural Language Processing*, Amsterdam, Benjamins Academic Publishers, pp. 209-278.

Mitchell, T. (1997), *Machine Learning*, McGraw-Hill.

Quinlan, J.R. 'Induction of decision trees', *Machine Learning.* Vol. 1, pp. 81-106.

Salton, G. (1980), 'Automatic term class construction using relevance: A summary of work in automatic pseudo-classification', *Information Processing and Management*, Vol. 16.1, pp. 1-15.

Sanromán, B. (2003), *Semdntica, sintaxis y combinatoria léxica de los nombres de emoci6n en español*, PhD Thesis, Helsinki, University of Helsinki.

Vossen, P. (1998), *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Dordrecht, Kluwer Academic Publishers.

Wanner, L., Alonso, M., Martí, A. (2004), 'Enriching the Spanish WordNet with Collocations', in *Proceedings of the LREC*, Lisbon.