STATISTICAL MODELING AT THE SYNTAX-SEMANTICS INTERFACE: EXPLOITING AUTOMATICALLY INDUCED LEXICAL CLASSES EVALUATED THROUGH VARIATIONAL BAYESIAN INFERENCE

A Thesis
Presented to
The Academic Faculty

By

Jonathan Ben Kamp
4063783

In Partial Fulfillment
of the Requirements for the Degree
Research Master of Arts
in
Linguistics

Utrecht University

August 2019

STATISTICAL MODELING AT THE SYNTAX-SEMANTICS INTERFACE: EXPLOITING AUTOMATICALLY INDUCED LEXICAL CLASSES EVALUATED THROUGH VARIATIONAL BAYESIAN INFERENCE

Approved by:

Dr. Tejaswini Deoskar
(Supervisor)
Department of Languages, Literature and Communication;
Utrecht Institute of Linguistics OTS
Utrecht University

Prof. Dr. Yoad Winter
(2^{nd} evaluator)
Department of Languages, Literature and Communication;
Utrecht Institute of Linguistics OTS
Utrecht University

Date Approved: August 29, 2019

***

A Gerrit e Giuseppe (†autunno 2018)

ACKNOWLEDGEMENTS

Thank you, Tejaswini, for introducing me to the computational syntax-semantics interface, and for keeping me on track. Thank you, Marijana and Henriette, and all professors involved, for the teaching and organization of this master. Thank you, Dylan and Gianluca, for the professional collaborations, fruitful insights, and friendship during this long way. Thank you, *basement* and *Delft-library-study-group*, for studying together and rounding up our theses. Thank you, Ombretta, for challenging me every day. Thank you, mamma, papà, Iris, and friends, for being there.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

So far, the task of automatic verb classification has been widely explored through supervised as well as unsupervised machine learning techniques, based on syntactic and semantic features, and strictly related to argument structure theory and Levin (1993)'s verb classes. In the present study we go a step further than the previous research in this field (e.g. Lapata and Brew, 2004, Merlo and Stevenson, 2001, or Sun and Korhonen, 2009) by using automatically induced verb classes not as a goal, but rather as a starting point for a lexicon induction experiment for individual verbs. Inspired by Rooth, Riezler, Prescher, Carroll, and Beil (1999), a first experiment involves a clustering process of verbs represented by co-occurrence vectors of argument nouns extracted from the subcategorization frames of transitive and intransitive verbs; from the resulting model, a second experiment shows that lexicons of argument nouns for fixed verbs can be created by re-estimating the nouns' absolute frequencies with respect to the same verb, modified by cluster-related probabilities from the model. Apart from being relatively simple statistical inference steps, the relevance of this study is also determined by the detailed and combined evaluation system used for model selection, including a Pseudo-Disambiguation task, in-depth cluster metrics, and a Variational Bayes Gaussian Mixture. It was found that argument selectional preference is a good indicator of verb classes, especially for the data set that included verbs of the alternation in which the object of the transitive is the subject of the intransitive. Moreover, through the support of a quantitative, WordNet-based method, it was shown that such classes are relatively little levinian. Future research could be directed to the exploration of adjunct slots, as well as an extension of the evaluation architecture to other clustering tasks within NLP.

CHAPTER 1
INTRODUCTION

Understanding the meaning of key lexical elements in a sentence can help to process *what is going on*, e.g. what situation is depicted, what actions are performed, what elements contribute to it and in what manner. In particular, the verbs in the sentence are highly informative about the latter aspects if compared, for example, to function words (e.g. articles) but also to nouns or adjectives. The study of the verb's semantics becomes, therefore, crucial to the overall understanding of the sentence.

Two of the of the foremost fields that attempt to explain the verb from this point of view are syntax and semantics. Precisely, cues from these fields contribute to the explanation of the verb's lexical semantics, based on its relation to its arguments in different forms. For example, it is assumed (Pustejovsky, 2016) that there are syntactical restrictions regarding the category of arguments that a verb allows in its structure, although they are not sufficient for a total explanation of its behavior. In fact, only in combination with semantic features, such as the lexical type of the required arguments, this can be achieved to a larger extent. But besides restrictions, an indicator of lexical semantics both in theory and in computational work (Sun & Korhonen, 2009) is selectional *preference*. This concept entails the affinity of a verb with different lexical elements or groups thereof as its arguments, in the way that they are distributed in natural language.

Driven by linguistic theory at the syntax-semantics interface and grounded in transparent statistical evidence, I approach the lexical and compositional meaning of the verb by an argument slot labeling experiment. Specifically, I propose an improved technique of automatic induction of slot annotation for subcategorization frames, inspired by Rooth et al. (1999). They showed how an EM-based clustering approach directed on an automatic slot labeling experiment yielded surprisingly well-performing results with relatively simple statistical inference steps. Compared to Erk (2007), for example, Rooth et al. (1999) trained a model that is better able to deal with low frequency verbs. This higher coverage, resulting in the property of estimating a lexicon for all the verbs in the corpus, makes of Rooth et al. (1999) the preferred baseline.

As Zapirain, Agirre, Màrquez, and Surdeanu (2010) neatly resume it, the concept of *semantic role labeling* refers to the operation of extracting simple event structures from language so to identify who did what to whom, when and where. Normally, this process has two steps: first, through syntactic queries and pre-processing, candidate predicate frames are extracted from a source corpus; second, the lexical features of the extracted elements are employed as statistical indicators for classification. In our case, the classification process is carried out by means of an unsupervised clustering technique, which is the general method that is used to divide unlabeled data into a given number of groups. The fact that the data samples do not have a class label *a priori* favors an unbiased classification of verbs that may even differ from linguistic theories. The fitted models are then used for a lexicon induction experiment for individual verbs, by means of re-estimation steps based on the conditional noun-verb probabilities given by the model.

1.1 Research Questions, Contributions, and Theoretical Relevance

The present study aims to carry out two experiments by applying up-to-date computational techniques and by motivating the methodological choices with strong explanations, both linguistically and data-driven. The relevance of performing such a research originates from the question of what verbal lexical semantics is and how it can be divided into more fine-grained particles that contribute to a higher meaning representation. In this sense, it is theoretically relevant to frame the semantic property of selectional preference in the role of indicator of a lexical class for the slots of an argument configuration.

In brief, the problems in previous work (Rooth et al., 1999; Erk, 2007) that I address can be summarized as follows: 1) the number of components in which the data is clustered, i.e. one of the most important parameters, is only partially accounted for; 2) the nature and quantity of the to be clustered verbs, together with their related co-occurrence nouns, is not defined precisely; 3) the cluster evaluation techniques are limited to one accuracy measure, without taking into account other statistical aspects of the different models. By acknowledging said limits, the following general research question naturally arises:

– "To what extent can the statistical indicator of selectional preference alone, based on a simple clustering algorithm and without the support of external linguistic resources (taxonomies or dictionaries), lead to the formation of semantically coherent classes that can be used for automatic slot labeling?"

From this question, several investigation lines follow. In particular, I will explore ways to render the clustering process decisions motivated by the data and evaluate them accordingly. Moreover, within this improved approach I make use of more advanced feature extraction and data pre-processing techniques than those that were used in Rooth et al. (1999). Choices for such methods and techniques partially derive from more modern resources available nowadays with respect to twenty years ago. On the other hand, different applications are used, such as a more accurate syntactic parser and specific dimensionality reduction techniques, as well as a refined clustering algorithm and cluster evaluation methods. It must be noted, however, that this research is not aiming at a state-of-the-art performance. Instead, it will be relying on a responsible methodology with up-to-date tools. One of the goals, in fact, is to create an evaluation system composed by the combination of different metrics, that can be directed to other tasks in computational linguistics as well. Concretely, I make use of metrics that are standardly used for the evaluation of EM-based Gaussian Mixture Models, namely the Aikake's Information Criterion, the Bayesian Information Criterion, and the Silhouette Coefficient. Furthermore, two important contributions of this thesis are the use of Variational Bayes Gaussian Mixture Models for evaluation, precisely for the estimation of the number of clusters to be initialized; secondly, a quantitative evaluation of the induced verb classes by the models based on, but different from, Sun and Korhonen (2009), which is necessary to evaluate the semantic coherency of classes in the light of linguistic theory (Levin, 1993). Finally, a Pseudo-Disambiguation task based on Pereira, Tishby, and Lee (1993) and Rooth et al. (1999) measures the generalization power of the models for which the optimal number of clusters is yet computed.

The options concerning the data were expanded to three sets. Not only the transitive and intransitive verbs of the BNC corpus Leech (1992) were considered (as in Rooth et al., 1999), but also two smaller samples. One of these takes into account the 500 most frequent verbs to see what effect the low frequency occurrences have on the generalization performance of the models; the other subset includes verbs that were classified by Levin

(1993) as belonging to the specific alternation where the object of the transitive is the subject of the intransitive form of the same verb. The latter data set could lead to more accurate clusters. The details of such an assumption are further explained in §3.2.5.3. Moreover, apart from EM-based clustering also used by Rooth et al. (1999), I adopt two other techniques: K-means and Spectral Clustering. This choice derives from the question whether Gaussian Mixture Models apply well to the linguistic data that is used, or that different algorithms may be preferred for future research.

1.2 Thesis Outline

This thesis is structured as follows. In Chapter 2, I present the linguistic and statistical theory behind the approach that I embrace. In this sense, there is an introduction to the study of verbal semantics, which leads to the essentials of distributional semantics and to previous work in automatic verb class inference from data. Chapter 2 concludes with the motivation of the present research in light of the background outlined in the same chapter. Then, Chapter 3 describes the data that is used, starting from a source corpus and ending up with three different, pre-processed data sets, including descriptions of the tools adopted to perform transformations of the data from one form to another. The first experiment is reported in Chapter 4, and includes how the models were fitted and evaluated, whereas Chapter 5 presents the second experiment, namely the semantic slot labeling. The results of Chapter 4 and 5 are further discussed in Chapter 6, in which we also encounter ideas for future work and a conclusion.

CHAPTER 2
THEORETICAL BACKGROUND AND RELATED WORK

In this section, the most relevant concepts and theories to the field of computational verbal semantics are introduced. Being the topic of this thesis a combination of theoretical linguistics and computational methods, the relevant background information consists of intermixed subsections, driven by the interdisciplinary perspectives on the field of computational lexical semantics. To begin with, §2.1 reports an overview of the theory of argument structure, selectional restritions, and selectional preference, and introduces Levin (1993), a work in which verb classes were manually drawn based on syntactic and semantic cues, as well as human-like intuition. §2.2.2 shows how similar verb classes can be automatically induced, leading us through previous work in the field, up to the motivation for the approach taken in the present research.

2.1 Verbal Semantics

Generally, the manner in which the study of semantics is approached, is twofold. On the one side of literature we encounter the branch of compositional semantics, which analyses meaning as deriving from the composition of elements, i.e. the manner in which single lexical elements determine the meaning of a phrase by the way that they are organized together. The other major subfield that can be identified, is the domain of lexical semantics, which focuses instead on the meaning of individual words. In the present thesis, the central linguistic problem is approached by the perspective of both lexical and compositional semantics.

Despite of the lexical semantics of content words (especially nouns) having been widely explored, and explained to a certain extent (for an introduction to one of the main challenges in this domain, i.e. polysemy, see Pustejovsky and Boguraev, 1997) the semantic interpretability of phrases and sentences still is problematic due to insufficient understanding of the lexical meaning of its individual components. Key components in this sense are function words (although, intrinsically, they carry more compositional than lexical meaning) and verbs. The verb is of particular interest since its lexical semantics stands central to the understanding of other elements of the same sentence, and to the semantics of the sentence in its entirety. Intuitively, the verb provides important information about the state, action, or event that is represented. Arguably, the verb's compositional and lexical semantics are partially influenced by syntax. According to a description proposed by Hackl (2013), related to the study of semantic cues and syntactic constructions that influence one another interchangeably and systematically, we find ourselves at the syntax-semantics interface. A key concept in this domain is that of argument structure, which is used to refer to a syntactic configuration produced by a lexeme, with associated semantic components (Hale & Keyser, 1998).

The different ways in which a verb combines with other lexical elements in order to form a grammatical sentence has also been widely studied. We may describe the different syntactic and semantic constructions of a verb as we encounter it in natural language as the verb's *behavior*. An intuitive action to take for a better understanding of a certain phenomenon of interest, is to create classes or categories so to group together its components. The rationale

behind this approach is that all members of a class share a similar pattern of features, which were established beforehand by the experimenter. In our case, the members are the English verbs, which have to be classified based on the subcomponents of their behavior. In the traditional literature of this domain, many have tried to explain the verb according to the linguistic subcomponents of the verb's structure and meaning. A representative concept in this context is that of θ-role theory. The main intuition is that a verb takes specific (types of) arguments, i.e. lexical items, that are required to fill the available thematic slots in the verb's frame. The idea of thematic relations was firstly introduced to the research community by Gruber (1965) and by Fillmore (1968) (who referred to this concept as *case relations*), and were then widely developed by Jackendoff (1972). What we may call a traditional system that comprehends a reduced set of discrete roles (e.g. Agent, Patient, Experiencer, Source) was taken as a mere starting point and was later exploited giving rise to diverse solutions to the problem of argument selection, i.e. what the founding principles are that determine the mapping between a θ-role and a grammatical relation. To give an example, Dowty (1989, 1991) places thematic roles in either a PROTO-AGENT class or a PROTO-PATIENT class, which are consequently split up in a hierarchical fashion into more fine-grained roles. The enormous quantity of roles that emerge from this does perhaps reflect the true nature of natural language, but at the same time lacks of generalization power in favor of a parametric framework. In order to overcome the latter barrier, Reinhart (2000, 2003) proposed the Theta System, according to which thematic roles can be reduced and translated into combinations of binary {+ or -} features, although pioneers of such boolean selectional restrictions were Katz and Fodor (1963). From a computational perspective this may be seen as an efficient system, since a consistent set of configurations is required that has as less rules and features as possible to successfully represent complex structures of meaning. But even with a low-level feature system like Reinhart (2000, 2003)'s, the problem in computational tasks remains that annotated data is expensive to construct. A more detailed overview of Reinhart (2000, 2003) is offered later in this section.

Moreover, we must bear in mind that there exist arguments against the presence of a syntax-semantics interface. So counter-argues Ravin (1990), who claims that syntax and semantics are in fact independent from each other, and that the θ-role theory in semantics is invalid. However, as computational work has demonstrated (see Sun and Korhonen, 2009 in §2.2.2), syntactic and semantic cues in combinations can truly be statistical indicators, which is reason enough not to deviate from the aforementioned theoretical perspectives in the present section.

From what we discussed, we can deduce that a central theoretical problem concerning the semantics of the verb is argument selection: can verbal meaning be defined by some selectional constraints that are intrinsic of the verb? But also: to what extent can verbs that have similar selectional constraints be grouped together into classes with consistent or uniform lexical meaning?

Pustejovsky (2016) refers to the process of accessing lexical information through syntactic and semantic operations as *selection*. This entails a salient lexical property of the verb, being the manner in which it is inserted in a phrase, regulated by syntactic and semantic boundaries. For instance, based on syntactic properties, a verb may allow one argument (intransitive), two arguments (transitive) or three arguments (ditransitive). In simplistic terms, these are the subject, direct object and indirect object of a verb, as can be observed in Example (1) for the verbs *cry*, *stub*, and *give*:

1. (a) Tom *cries*.

(b) Luke *stubbed* his left foot.

(c) John *gave* him a lesson.

Following Pustejovsky (2016), the argument structure of the verbs in Example (1) can be formalized as follows:

2. (a) **cry**(arg_1)

   (b) **stub**(arg_1, arg_2)

   (c) **give**(arg_1, arg_2, arg_3)

It is worth underlying that the number of arguments of a function such as each of the three verbs in Example (2), is the number that is required to render the predicate complete in terms of its available argument slots. In natural language, *complete* would thus indicate that the expression that contains or constitutes the predicate is grammatically *well-formed*. We refer to the necessary number of arguments of a verb as its *valency* or *valence*, what — in logic and mathematics — is called the *arity* of a function. Therefore, phrases that behave as optional arguments of a verb (i.e. *adjuncts*), are not considered as a requirement for the grammaticality of an expression. Example (3a) shows the transitive verb *pass* with two mandatory arguments *Mary* and *Jack* and one optional adjunct *the salt*. In fact, the argument structure of a transitive verb is formally different from a ditransitive verb (compare 2c and 3b).

3. (a) Mary *passed* the salt to Jack.

   (b) **pass**(arg_1, arg_2)

Moreover, the property of argument selection can be regarded as pointing into two similar directions: *selectional restrictions* versus *selectional preference*. On the one hand, selectional restrictions prevent the predicate from accepting more or less arguments than its valence allows. A violation of this restriction would result in ill-formed expressions. To give an idea, Example (4a) takes two arguments instead of one, (4b) takes one instead of two, and (4c) takes two instead of three:

4. (a) *Tom *cried* the girl.

   (b) *Luke *stubbed*.

   (c) *John *gave* him.

Apart from the valency restrictions, selectional constraints also control for the syntactic category of the argument, i.e. the type of phrase a predicate requires, and the number, i.e. the singular or plural form (Pustejovsky, 2016). These two aspects are represented in Example (5), where the intransitive verb *meet* is taken as an example of how only a plural noun phrase is allowed as arg_1.

5. (a) The guys *met*.

   (b) **meet**$(arg_1[cat : NP, plural : +])$

A third restriction originates from a semantic feature of the lexicon, namely animacy. Example (6) illustrates how the violation of this binary feature causes the sentence to be ill-formed.

6. (a) *My shoe *giggles*.
   (b) **giggle**($arg_1[cat : NP, animacy : +]$)

Note that a verb may not have any restriction at all regarding animacy or number, making the explicit indication of it redundant. For instance, in Example (6b) a number restriction is omitted since it does not apply to the verb *giggle*.

Nevertheless, it is clear that such a low number of restrictions is insufficient to account for the full spectrum of lexical aspects of the argument, including their semantic role in the predicate and sentence. This brings us back to Reinhart (2003), who's system of binary features could offer a deeper insight of the lexical problem in question. Her proposal is relatively simple, transparent and organised, making it thereby directly accessible to computational approaches, especially because of its quantitative, boolean nature. The baseline of her system consists of two binary features, namely $\pm c$ and $\pm m$. c stands for *Cause change*, whereas m indicates the presence of *Mental state* in the lexical expression of the argument. The various combinations of these features together give rise to a series of clusters (not to be confused with the term *cluster* in the next chapters, which indicates a component of a clustering model), which point at specific, by context determined theta-roles, inspired by Dowty (1991). In Example (7) it can be observed how different clusters can be formed and related to a proto-role on the right:

7. (a) $[+c + m]$ - agent
   (b) $[+c - m]$ - instrument
   (c) $[-c + m]$ - experiencer
   (d) $[-c - m]$ - theme / patient
   (e) $[+c]$ - cause
   (f) $[+m]$ - sentient
   (g) $[-m]$ - subject matter / locative source (Typically Oblique)
   (h) $[-c]$ - goal / benefactor (Typically Dative (or PP))
   (i) [ ] - arbitrary

The Theta System provides combinations of $\pm c$ and $\pm m$, only $\pm c$ or $\pm m$, or an empty, arbitrary role. In this manner, the argument structure of a verb can be formalized as follows:

8. $V([+c], [-c - m])$, where V= *open*, *break*, *melt*, etc.
   (a) The wind / Max / the key *opened* the door
   (b) The storm / Max / the stone *broke* the window.
   (c) The heat / Max / the candle *melted* the ice.

The labeling of the arguments with similar feature clusters seems more precise than the selectional restrictions that we observed in example (6), i.e. the sole syntactic category and animacy aspect.

As previously discussed in this section, besides restrictions there also exists and aspect of argument selection named selectional *preference*. The idea is that a verb not only occurs with specific lexical elements in its argument slots with a different frequency compared to other verbs and other nouns (i.e. *lexical preference*), but also that these lexical elements

can be grouped together into a higher taxonomic rank. For example, the transitive verb *eat*, for the sake of explanation, has a lexical preference that is highest for *he* as arg_1 and *apple* as arg_2, and lowest for, say, *stone* and *roof* as the same two arguments. Instead, in terms of selectional preference, larger lexical groups could be formed and assigned to the argument slots of the verb by abstracting over the individual lexemes and extracting a set of essential, discrete features. These higher lexical classes can be described in the same fashion as the feature clusters by Reinhart (2003). A way of representing such preferences will be explained in §2.2.1.

As for most scientific descriptions of a certain phenomenon, splitting the latter up into groups or classes is a intuitive way of representing it, as well as a control used to verify that the features that are considered indicators of a certain phenomenon are able to discriminate the different examples in the data. In other words, creating groups of instances of the data is a way to test whether observations of apparent patterns in the data can lead to valid generalizations. In the case of the verb, the question of what lexical semantic traits — coarse-grained or atomic — define its meaning, is still pending. In §2.1.1, I present a piece of manual verb classification research that also inspired computational approaches.

2.1.1 Levin's Verb Classes

A major work in verb categorization has been proposed by Levin (1993). Her study can be regarded as the theoretical and intuition-based foundation of a large slice of current computational research. Despite the fact that the verb classes that Levin (1993) defines, have validity for the English language only, this categorization illustrates the way in which semantic and syntactic features in combination with one another can lead to classes of verb senses that are more or less consistent with human-like intuition, and that are semantically *coherent*. However, since no automatic verb classification task has achieved an accuracy that is substantially higher than 80% (Sun & Korhonen, 2009) (see 2.2.2 for a more detailed follow-up) even with sophisticated linguistic features, it is clear that the fine-grained semantic particles that the verb may be composed of, are yet far from discovered or explained.

The work of Levin (1993) has two parts. The first part describes a list of *diathesis alternations*, followed by a set of verb classes in part two that are partially based on the former. Diathesis alternations are *"alternations in the expressions of arguments, sometimes accompanied by changes of meaning"* (Levin, 1993). A specific verb may participate in such an alternation, for example *break* in the following two sentences:

9. (a) The girl *broke* the window.
   (b) The window *broke*.

The example expressions in (9) represent a case of the *causative-incohative* alternation, where the verb *break* allows two different argument structures to express the same action of the window breaking, although with a small difference in meaning concerning the cause of the action. Similarly, the *locative* alternation allows two different argument configurations, made possible by a use of different prepositions, too:

10. (a) Sharon *sprayed* water on the plants.
    (b) Sharon *sprayed* the plants with water.

At the same time, semantically similar verbs to *spray*, like *cover* and *pour*, do not display the a positive grammatical judgment as for (10):

11. (a) *Monica *covered* a blanket on the baby.
    (b) Monica *covered* the baby with a blanket.

12. (a) Carla *poured* lemonade into the pitcher.
    (b) *Carla *poured* the pitcher with lemonade.

Through the extended examples and descriptions of diathesis alternations in English, Levin (1993) shows how the low-level argument structure properties do not suffice for an explanation of the verb's behavior. In fact, the speaker's natural intuitions and knowledge about which encodings of the verb are or are not allowed by grammar is highly discriminative, and may therefore reside also outside the lexical expression of a word. We commonly refer to this difficult to represent, and hardly to encode concept as *world knowledge*.

The impact of Levin (1993) has been of such an importance that her framework of sense-grounded verb classes has been extended to a digital level throughout the years. The result thereof is VerbNet (Kipper-Schuler, 2005)[1], a collection of fine- and coarse-grained classes inspired by Levin (1993) (see Kipper-Schuler, Korhonen, Ryant, and Palmer, 2006 for information about such extensions) with mappings to the taxonomic resource WordNet (Miller, 1995), but also to PropBank (Kingsbury & Palmer, 2002) and FrameNet (Baker, Fillmore, & Lowe, 1998).

In light of the linguistic theories on theta relations that I just described, we are now familiar with the theoretical context that is relevant to the present thesis. In order to approach verbal semantics with as few theoretical assumptions as possible, statistical modeling at the lexical level may seem an adequate solution. In the next section (§2.2.1) an overview is presented of distributional modeling.

2.2 Verb Classes from Data

2.2.1 Distributional Semantics in a Nutshell

At the background of the approach and computational implementation that the present study adopts, we find the distributional hypothesis (DH). Although I deviate from the pure DH, I consider a short introduction to such a perspective in place.

According to the DH, the meaning of a lexeme can be estimated by the words that it goes together with, i.e. by its context. For example, given an unknown word *besariz* and several contexts in which it occurs, we can attempt a good guess of what the word in question refers to:

13. (a) I love using a *besariz*, in fact I have two of them.
    (b) The chef was cooking Dutch paella with a *besariz*.
    (c) A pleasant metallic sound is produced when you flip a *besariz* and tick on it.
    (d) At Ikea, *besarizes* are very, very cheap.

[1] https://verbs.colorado.edu/verbnet/

Given the knowledge of the reader regarding kitchen utensils, they would probably interpret *besariz* as a type of cooking pan or a similar object. In fact, the word *pan* could naturally replace the word *besariz* in all the four proposed contexts, whereas the word *water*, for instance, would not fit in any of them. We refer to distribution of a lexeme as the manner in which it is distributed in natural language (i.e. in which contexts it occurs). This is directly related to the properties of argument structure given by Pustejovsky (2016) who, regarding the topic of argument selection in §2.1, claims that *"one of the most important properties of a verb is an encoding of what phrases it can appear with in the language"*. Nevertheless, a distinction must be made between the general approach of distributional semantics and ours. In fact, a computational implementation of a context usually refers to its representation in the form of a vector, which can be binary when the pure absence or presence is relevant for the task, or count-based when the distribution of the context co-occurrences for a given word is requested. The vectorized representation of the context is then used as an approximation of the word's lexical semantics. As we will see in §3.2, however, the direct relationship between vector and meaning is not of our interest. Instead, the vectors will encode the lexical distribution of the arguments over the verbs in question in order to form semantically coherent verb classes.

The origins of the DH can be encountered in Harris (1954), who argued that *"difference of meaning correlates with difference of distribution"*. From that moment on, distributionalism found its way and developed itself into several fields, such as Psychology (Osgood, 1952) and Linguistics. According to Lenci (2018), however, a distributional framework was seen as an alternative to the more traditional formal and logical approaches to semantics only after its exploit in information retrieval, caused by a raise in popularity of statistical NLP in the nineties.

In Linguistics, a distributional semantics approach has been applied to several tasks. One of the first, and perhaps one of the most intuitive suggestions, came with Garvin (1962). He acknowledged the limits of the approaches to linguistic analyses at that time, due to restricted rule- and dictionary-based methods in the fields of machine translation and information retrieval (IR). In the same field of IR, Salton, Wong, and Yang (1975) proposed a more concrete vector space model, which was able to encode and represent semantic similarity relatively successfully and as an alternative to formal semantics. Also compared to taxonomic resources, (e.g. WordNet by Miller, 1995) which are based on human-like perceptions of the world's semantic relations, distributional models offer a more theory neutral option that is almost solely based on statistics. It comes natural, then, to prefer a statistical approach as little prior assumptions must be made, so that relatively unbiased linguistic patterns (that may even contrast human intuition) can be discovered.

2.2.2 Automatic Verb Classification

What follows, is a short overview of studies that treat the topic of automatic verb classification, which forms an experimental foundation of the present approach. The idea behind this overview is that it would help gain an better understanding of where the present study broadly bases its techniques on, and also how it goes a step further, as explained in §2.2.5. Some of the studies described adopt a supervised machine learning method, meaning that data examples (e.g. verbs) are labeled with a class prior to classification. In unsupervised techniques (usually clustering-based), only the potential features as statistical indicators are defined, without prior labeling. Supervised methods are overall more accurate than unsupervised methods, but are also biased by the yet established label.

2.2.2.1 Merlo and Stevenson (2001): Automatic verb classification based on statistical distributions of argument structure

Merlo and Stevenson (2001) report on a series of supervised learning algorithms to classify three types of optionally intransitive verbs based on their argument structure: unergatives, unaccusatives and object-drop verbs. They achieved a 69,8% accuracy score on this task. The features that were used for training are all linguistically motivated and consist of the following: transitivity, causativity, animacy, and two additional syntactic features being the use of passive or active voice and the use of the past participle or simple past POS tag. The data was collected from two corpora: an automatically tagged, combined corpus (primarily Wall Street Journal (WSJ)) of 65 million words, and an automatically parsed corpus of 29 million words (a subset of the WSJ). As for the experimental methodology, Merlo and Stevenson (2001) investigated several supervised learning algorithms (decision tree induction, rule learning, and two types of neural networks), finding an approximately equal performance rate for all classifiers, For testing, 10-fold cross-validation was used in one run of experiments, whereas a single hold-out training and testing approach ($N-1$-fold cross-validation) was used in the other.

2.2.2.2 Lapata and Brew (2004): Verb class disambiguation using informative priors

Lapata and Brew (2004) approached the problem of verb class disambiguation building informative priors. They showed how to train and use a probabilistic version of Levin (1993)'s classification, taking as input a partially parsed corpus and returning a probability distribution over the available verb classes for each combination of a verb and its syntactic frame. The assumption is that, in a given frame, the choice of a class for a polysemous verb is considered as maximizing the joint probability *p(class,frame,verb)*. They showed that subcategorization information acquired automatically from the BNC corpus could lead to important cues for verb sense disambiguation. As only the most preferred class is predicted, the result of Lapata and Brew (2004)'s work is not more (but also not less) than an informative prior for a complement a verb classification system. Since the model does not take into account selectional restrictions, discourse, or pragmatic information, the prior yields especially useful information when knowledge to the former aspects is not accessible.

2.2.2.3 Li and Brew (2008): Which are the best features for automatic verb classification

One of the best performing supervised methods in terms of F-measure, is that of Li and Brew (2008). They used a Bayesian Multinomial Regression for classification, training on features that were extracted from the large Gigaword corpus (collection of samples of recent newswire text data). They regarded this log-linear modeling framework, which is similar to Maximum Entropy, as the most appropriate algorithm for automatic verb class induction, outperforming SVMs. Specifically, it works efficiently with large numbers of features and extremely sparsely populated matrices (Li & Brew, 2008). Their main finding is that subcategorization frames are not the most effective features for this task; instead, the suggestion is to use both syntactic and lexical information together as predictors. Combining these aspects can be done in multiple ways: dependency relations, co-occurrences, adapted co-occurrences, subcategorization frames + co-occurrences.

2.2.2.4 Sun and Korhonen (2009): Improving verb clustering with automatically acquired selectional preferences

As for unsupervised learning, Sun and Korhonen (2009) performed relatively well. They used a variation of Spectral Clustering, useful for high dimensional feature spaces, based on the MNCut algorithm, which was also the implementation choice of Brew and Schulte im Walde (2002). They tested a set of features that is similar to the one considered by Joanis, Stevenson, and James (2008), finding that the best combination of predictors consisted of subcategorization frames together with semantic cues. In order to evaluate their clustering method, Sun and Korhonen (2009) employed several measures: the first one consists of the modified purity score, a global measure which evaluates the mean precision of clusters, while the other one is a weighted accuracy score. By regarding the former as a precision indicator and the latter as recall score, a final, weighted F-measure was computed between the two.

2.2.3 Studies on Selectional Preference in Computational Semantics

The lack of annotated data but abundance of plain text corpora indicate the need of linguistic features that are directly extractable from texts on which to apply some automatic pre-processing, but no manual labeling. Resnik (1993, 1997) proved as first that the theoretical concept of selectional preferences (SPs), the typicality of arguments in relation to a specific predicate, can be a statistical indicator in an NLP task. Moreover, in computational linguistics, SPs have proven very useful for syntactic disambiguation (Hindle & Rooth, 1993), word sense disambiguation for nouns, verbs, and adjectives (McCarthy & Carroll, 2003), and semantic role labeling (apart from Rooth et al., 1999, also Gildea and Jurafsky, 2002, although the latter system was created to predict the thematic role for a word in a sentence and was depending on FrameNet, whereas the former did not make use of other resources). Concretely, Resnik created a language model that was capable of formalizing *selectional preference strength* of a predicate in terms of relative entropy (information theory; see Kullback and Leibler, 1951) and prior-posterior probability distributions over the classes:

$$S_R(p) = \Sigma_c Pr(c|p) \log \frac{Pr(c|p)}{Pr(c)} \tag{2.1}$$

where $Pr_R(c)$ is the prior distribution of a class c occurring as the argument in predicate-argument relation R. The idea of $S_R(p)$ is that it measures how much information predicate p provides about the latent class of its argument. The ground-truth classes adopted are WordNet synsets, which makes the model depending on an external taxonomic source and therefore different from the fully authomatic approach by Rooth et al. (1999)[2]. The SP of p for synset c is defined as the contribution of c to p's selectional preference strength $S_R(p)$:

$$A_R(p, c) = \frac{1}{S_R(p)} Pr(c|p) \log \frac{Pr(c|p)}{Pr(c)} \tag{2.2}$$

Relatively recent research on automatic induction of SPs and semantic role labeling comes with Erk (2007). She proposes a model using corpus-based semantic similarity metrics (e.g. cosine similarity and Lin et al., 1998's mutual information score), obtaining lower error rates than both Resnik's WordNet-based model and the EM-based clustering model based

[2]See Brockmann and Lapata (2003) for an evaluation and comparison of (WordNet-based) SP acquisition models

on Rooth et al. (1999), but with worse coverage, preventing to successfully deal with sparsity. Erk (2007)'s model does not rely on external lexical resources or manual annotations, and applies semantic role labeling following an existing collection of verb frames (FrameNet, Baker et al., 1998), learning different preferences for the different senses of a word. It is worth noticing that the EM-based model trained by Erk (2007) for the sake of comparison, differs from Rooth et al. (1999) in terms of accuracy, reporting accuracy scores of around 65%-69% and 80%, respectively. I will take said performances as a benchmark for the clustering results in Chapter 4. Similarly, Ó Séaghdha (2010) created a series of topic models for SP induction, based on Latent Dirichlet Allocation and evaluated on human plausibility judgements. These models obtain a high coverage accuracy, accounting for infrequent predicate occurrences.

A different path of research is the one approached by Bergsma, Lin, and Goebel (2008). They suggested learning selectional preferences in a discriminative way, by training a collection of Support Vector Machine classifiers to recognise what lexical elements are more or less likely used as arguments for a given predicate. However, the generation of semantic classes they performed, led to 3620 clusters, a number for which they did not account, apart from being relatively high independently from the size of the corpus data. The number of components will be a central methodological issue in the present study and will be explained more in depth in Chapter 4.

2.2.4 Rooth et. al (1999): Inducing a Semantically Annotated Lexicon via EM-Based Clustering

Compared to the studies described in §2.2.2, Rooth et al. (1999) use a similarly trained model as a starting point for a second step. Precisely, besides an initial verb classification process, an automatic induction phase is carried out to create lexicons of subject and direct object arguments for fixed transitive and intransitive verbs. The notable aspect of their approach is that the statistical methods used are relatively simple and do not rely on external resources such as WordNet or FrameNet, apart from a necessary source corpus. First, simple count vectors are created that register the distribution of specific nouns over the argument slots of transitive verbs (one subject slot, one direct object slot) and intransitive verbs (one subject slot). Thus, each verb will have a total of three co-occurrence vectors, one for each potential slot: transitive subject, transitive object, intransitive subject. These different uses of specific verbs with respect to a fixed lexicon of nouns Ln, i.e. the set of all nouns that occur as subject or direct object argument for the set of verbs in question, are clustered, giving rise to classes that share similar distributional patterns over Ln. The second step involves the creation of a lexicon of nouns for the argument slots of individual transitive and intransitive verbs. This is based on an adjusted frequency score, resulting from the re-estimation of the nouns' absolute frequencies for a fixed slot together with the probabilities of all nouns' frequencies extracted from the cluster in which the verb in question was classified.

2.2.5 A Combined Approach: Motivating the Present Research

Given the research in automatic verb classification seen in §2.2.2, one way to further explore that field is to find syntactic, semantic, or other types of linguistic features that may improve the performance of the classifier. Such an approach is theoretically interesting, since an atomization of believed meaning components into further fine-grained particles would not

only contribute to a more accurate computational model, but would also tell what syntactic and semantic cues are effectively discriminating for the meaning of two verbs, as well as their weight in a major meaning representation. In the end, however, it is questionable what an accuracy improvement of a few percentages of the verb classifiers contributes to the research area. Instead, taking a step further than only building a classifier model, by carrying out an NLP task on top of it, would provide a concrete contribution to the field. Inspired by Rooth et al. (1999) (which obtained more promising results than similar research as described in §2.2.3), the present research not only shows how verbs can be automatically grouped together, but also how the same model can be used in a second experiment to estimate a lexicon of arguments for individual verbs.

The choice for an unsupervised machine learning approach derives from the fact no annotation is required, and that patterns are automatically learned from this unlabeled data, contrarily to the studies described in §2.2.2.1, §2.2.2.2, and §2.2.2.3. This is desirable, since no strong linguistic framework of verb classes that could bias the experiments is assumed beforehand.

CHAPTER 3
THE DATA

In this chapter, I explain the choices that were made with respect to the original data source and the transformations thereof, up to the data sets that are used for the clustering experiments in Chapter 4 and 5. First of all, I introduce the source corpus in §3.1, motivating the choice of basing the experiments on the corpus in question rather than on a different one, and reporting some descriptive statistics that are directly related to it. In a second phase, starting from the raw corpus data, I report how elaborated feature extraction and data pre-processing steps have been brought to completion, including parsing details in §3.2.2 and a dimensionality reduction technique in §3.2.4. The result of these data preparation procedures applied on the initial corpus yield three data sets, a description of which is given in §3.2.5. These data sets are formed in such a way that they are ready to be fed to the machine learning algorithms in Chapter 4.

3.1 The Corpus

The data was gathered from the same linguistic source that was used in Rooth et al. (1999), being the British National Corpus (BNC) Leech (1992). Although the real usage of a particular language should be reflected by the lexical distribution of any similar corpora of sufficiently large size, using the approximately the same corpus directly entails the preservation of a meaningful comparison of the results. Besides, the BNC was also used as data source in Erk (2007)'s strictly related work, as described in §2.2.3.

The BNC is a collection of British English textual records of written and spoken data (90% and 10%, respectively). It is a synchronic corpus and displays language use of the late 20th century, which makes it relatively representative of current British English. Furthermore, the BNC can be regarded as a balanced corpus given its composition of different source types, such as newspapers, journals, periodicals, fiction books, context-controlled speech and natural speech (Leech, 1992).

The total size of the BNC counts a 100 million words. In order to access the XML version of the BNC[1], the NLTK (version 3.4) (Loper & Bird, 2002) corpus reader tool was used.

3.2 Feature Extraction and Data Pre-processing

3.2.1 Hardware and Software

A substantial part of this research was technical in nature. Usable data structures had to be created out of an initial corpus, and had later to be used for training a series of models. Finally, models had to be tested and the results had to be visualized. For these purposes, computational tools were used. In this section, I introduce the hardware on which computations were performed across the entire study, as well as the main software specifications. Besides, I provide details about the packages that were used for the data preparation and

[1] http://www.ota.ox.ac.uk/desc/2554, accessed on April 18th, 2019

manipulation, whereas further technical information about the two experiments is given in §4.4 and §5.1.

The laptop on which the data was handled is a Lenovo ideapad 330S-15IKB, with a processor Intel Core i7-8550U (8th Gen), 8,00 GB of RAM, CPU @ 1.80GHz, 2001 Mhz, and running on Microsoft Windows 10 Home. I used Python 3.7.3 through Spyder 3.3.5 computing environment. The most relevant package that I used for data pre-processing and storage is SciPy (Jones, Oliphant, Peterson, et al., 2001–) 1.2.1, which allowed me to handle highly sparse data structures (from dictionary-like structures to *scipy.sparse.dok*, to *scipy.sparse.csr*)[2]. The Scikit-learn library (Pedregosa et al., 2011)[3] was widely exploited for both data preparation and modeling.

With respect to computation times, worth reporting is the fact that the parsing process described in §3.2.2 lasted approximately 3 to 4 hours, while the creation of sparse data structures in §3.2.3 took between 30 and 60 minutes. Minor data transformations did not exceed 20 minutes of time.

3.2.2 Parsing

Within the feature extraction phase, the first step is to parse the sentences of the BNC. The NLTK corpus reader already provides the boundaries between sentences and part-of-speech tags for the individual words. In this way, full sentences that include at least a verb are easily extracted, and segments without a predicate (e.g. titles) are ignored. The resulting potential sentences with at least a transitive or intransitive verb were then parsed in order to extract the verb and its noun argument(s). For this task I used EasyCCG, an A*-search-based, state of the art parser[4] (Lewis & Steedman, 2014), to convert unstemmed sentences of the BNC into syntactically and semantically interpretable data. Rooth et al. (1999) used a head-lexicalized context-free grammar parser (Carroll & Rooth, 1998) that reaches performances of 79% precision and 75% recall. The EasyCCG parser, instead, was developed 15 years later and benefits from computational advances and improved linguistic theories in this temporal window, obtaining an accuracy of above 98%. Unlike dependency parsers like Stanford parser (D. Chen & Manning, 2014), EasyCCG makes a distinction between arguments and adjuncts. Hence we decided to use it, in addition to overall accuracy. Precisely, Combinatory Categorial Grammar (CCG) generates constituency-based structures based on logic combinatory power, and is able to represent linguistic structures well, such as argument-structure theory and semantic set-theory. The first linguistic arguments for basing the grammar on combinators were put forth by Steedman (1987, 2000) and Szabolcsi (1992). In short, CCG theory labels words in a sentence with simple POS-tags or, instead, with a more complex combinator that tells how the word that it is assigned to combines with other, adjacent words in a sentence into a higher phrase. As we can observe in Figure 3.1 and 3.2, individual elements are combined in incrementally larger combinations up to the declarative sentence *S[dcl]*. In Figure 3.1, a transitive sentence is represented, whereas in Figure 3.2 an intransitive sentence is displayed. The parsing system works such as so to assign a specific combinatory label to the verb, depending on the manner in which it combines with its arguments in the sentence. In both example parses the verb is *twisted*, which takes in both cases an NP to the left (indicated by a backslash character), being the subject NP[nb] *The boy* and *The boy's ankle*. Next, the verb in the transitive sentence also

[2]https://docs.scipy.org/doc/scipy/reference/sparse.html, accessed on April 18th, 2019
[3]https://scikit-learn.org/stable/, accessed on April 18th, 2019
[4]http://homepages.inf.ed.ac.uk/s1049478/easyccg.html, accessed on April 18th, 2019

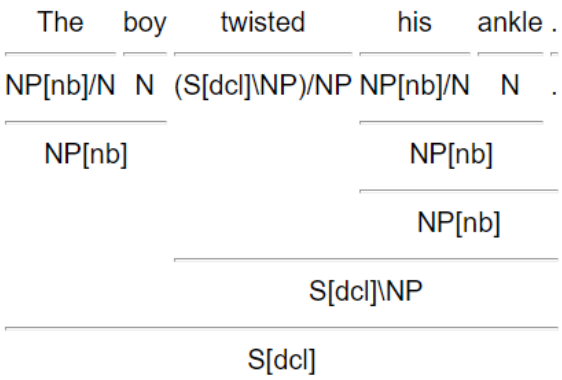

Figure 3.1: Example of CCG parse of a transitive sentence.

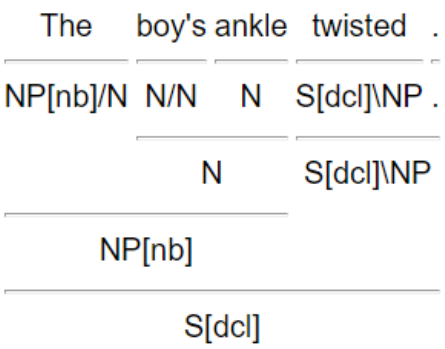

Figure 3.2: Example of CCG parse of an intransitive sentence.

takes an NP[nb] to the right (indicated by a forward slash character) as the object, which, in the case of Figure 3.1, is *his ankle*.

The settings of EasyCCG were left at default values, including the option of skipping sentences that are longer than 70 words to maintain a solid parsing accuracy, and therefore yielding a coverage of above 98.0% of all sentences.

3.2.3 Creating a Co-occurrence Matrix

From the parsed sentences of the BNC, the verbal instances that take as arguments only a subject NP (S[dcl]\NP) or both a subject NP and a direct object NP ((S[dcl]\NP)/NP) were extracted. This process resulted in the creation of 5.111.245 tuples, consisting of $(n1, v)$ pairs and $(n1, v, n2)$ triples. The frequency of extracted tuples summed up to 14.979.954 tokens. Prior to extraction, verbs and nouns were lower-cased, as well as lemmatized. Lemmatization of nouns was performed through the NLTK WordNet Lemmatizer[5] (Loper & Bird, 2002), whereas verbs were lemmatized with the Pattern lemmatizer[6] (Smedt & Daelemans, 2012) given a higher performance in terms of coverage (8.500 common English verbs). Although the latter tool lacks of applicability to uncommon verbal entries, this is not a problem for the feature vectors since the majority of them are hapax legomena or, in any case, extremely rare. In other words, a lemmatized hapax legomenon will weight the same

[5]https://www.nltk.org/_modules/nltk/stem/wordnet.html, accessed on April 18th, 2019
[6]https://www.clips.uantwerpen.be/pages/pattern-en, accessed on April 18th, 2019

as its unlemmatized counterpart. Conversely, we do not want to consider different inflected forms of a more common verb as to be clustered independently the one from another.

The resulting matrix counted 41.916 verb vectors in combination with the weights given by 113.674 noun-based co-occurrence feature vectors. The 41.916 verb vectors are the result of 13.972 verbal entries that are present in the active transitive or active intransitive structure in the BNC, each with the three possible slot options ".tr:s", ".tr:o", and ".intr:s" (abbreviations for "transitive:subject", "transitive:object", and "intransitive:subject"). As we will see in §3.2.5, the matrix described in the present section is the underlying data structure of the ALL data set and its subsets F500 and LEV. However, before being fed to the clustering process, further transformations of the data were applied. The first modification is normalization, performed sample-wise (i.e. verb-wise) and not feature-wise (i.e. noun-wise), with *l2* norm (Pedregosa et al., 2011). Normalizing is a form of scaling the data so that factors as absolute frequency do not affect the weight of a vector excessively. For example, a verb that occurs far more often than another may differ in frequency, but not in the distribution of the nouns over their argument structure; bringing the vector counts to the same scale overcomes the problem of different weightings due to absolute frequency. Moreover, the models that were created with normalization showed a clearly more equally distributed data over the classes, whereas unnormalized data caused one highly populated verb class with $> 95\%$ of the data, and mostly singleton clusters. Given these consequences, it can be inferred that such a skewed distribution is relatively uninformative, as discriminant patterns in the data are evidently missed by the algorithm.

In the next section, §3.2.4, a second, major process that was applied on the data is described. Herein I describe how SVD works, a form of dimensionality reduction, what settings are chosen, and a motivation for using technique in question.

3.2.4 Dimensionality Reduction: Singular Value Decomposition

Given the high dimensionality of the feature space, I applied truncated Singular Value Decomposition (SVD) through Scikit-learn (Pedregosa et al., 2011) to reduce the number of vectors of the term count matrix. Precisely, SVD is the matrix reduction algorithm in Latent Semantic Analysis (LSA), originally proposed by Deerwester, Dumais, Furnas, Landauer, and Harshman (1990) and used for a wide range of NLP applications (see Jurafsky and Martin, 2014 for examples). According to Lenci (2018), it is the most commonly used feature extraction technique in distributional semantics, not only because of its computationally efficiency, but also for the fact that it keeps the informativity of the dimensions intact with respect to the original, non-factorized input data. Specifically, SVD is a type of matrix factorization that has the effect of combining co-occurrence columns, so to give weight to the more informative or salient features in the data. Using Lenci (2018)'s notation and descriptions, SVD factorizes an $m \times n$ co-occurrence matrix M into the product of three other matrices, where $z = min(m, n)$:

$$M_{m \times n} = U_{m \times z} \Sigma_{z \times z} (V_{n \times z})^T \tag{3.1}$$

Here, $\Sigma_{z \times z}$ is a square diagonal matrix containing singular values that are sorted in descending order. The latent dimensions in the input data are represented by the columns of the matrices U and V, and are ordered by the amount of variance in the data that they respectively account for. Based on the choice of a parameter k that is given at the time of applying SVD, the first k singular values (i.e. the values in $\Sigma_{z \times z}$) and the first k singular

vectors (i.e. the columns in U and V) are kept, returning a final matrix.

A drawback of SVD is that the higher level abstractions produced by these feature combinations are, apart from hardly interpretable, mere approximations. However, the result is the best approximation of M that maintains the variation in the data. Concretely, a certain amount of information is lost during the matrix factorization, but this is counterbalanced by the improved weighting of informative patterns in the data. Standard values for the number of first dimensions to keep range approximately between 50 and 500. As we want to lose as little information as possible from the feature vectors, 500 is the preferred value for said parameter.

3.2.5 Data Sets for Clustering

The parsing, feature extraction and pre-processing steps have transformed the semi-raw corpus data into a form that is manageable for the unsupervised machine learning algorithms that will be adopted in Chapter 4. In the present section, I outline the three final data sets that result from the aforementioned processes. Note that the selection and filtering of verbs that give rise to the different data sets are applied before normalization and SVD, which means that the smaller data sets in §3.2.5.2 and §3.2.5.3 do not originate from the larger cleaned data set in §3.2.5.1, but are the result of a similar process starting from the extracted BNC tuples.

3.2.5.1 The ALL Data Set

The first data structure is the ALL data set. It directly derives from the extracted transitive and intransitive verbs of the BNC. The ALL data set includes, as anticipated in §3.2.3, 14.979.954 tokens, given by 13.972 verb types (41.916 vectors). SVD reduces the features to 500.

3.2.5.2 The F500 Data Set

The motivations for a second data set, F500, originate from the fact that the ALL data set is highly sparse, with 90% of the occurrences pointing to the 500-1000 most frequent verbs in the corpus. As a matter of comparison, I take the subset of transitive and intransitive cases of the 500 most frequent verbs into account when analysing the clustering results of all verbs in this specific alternation. F500 counts 13.405.110 tokens from tuple extraction, 500 verb types and 1.500 vectors. SVD reduces the features to 500. The F500 data set can be inspected in Appendix A.

3.2.5.3 The LEV Data Set

A third data set is created from the original corpus by considering only verbs that bear the property of belonging to the transitive alternation. In order to select such set of verbs, I exclusively considered the verbs described by Levin (1993) that display a transitive-intransitive alternation, and more precisely those showing the particular argument configuration where the object of the transitive is the subject of the intransitive. This group includes the *Middle Alternation*, the *Causative Alternations* — covering the *Causative-Inchoative Alternation*, the *Induced Action Alternation*, plus other cases — and the *Substance/Source Alternation*. This data set has the property of being potentially more robust than the previous two, despite the fact that it contains less verbs (both tokens and types) and that overall they

are relatively uncommon. The rationale behind this choice is this that clusters of the model will still be the result of a training phase based on three vectors per verbal entry, but with the peculiarity of being two of them, the *.intr:s* and the .tr.o, distributed more similarly to each other than the same vectors for the ALL and F500 data sets. LEV counts 448.250 tokens, 368 verb types and 1.104 vectors. SVD reduces the features to 500. The LEV data set can be inspected in Appendix A.

CHAPTER 4
EXPERIMENT 1

4.1 Task: Unsupervised Verb Classification

In this chapter, I present the first of the two experiments. The first experiment consists in the creation of a Gaussian Mixture Model based the verb vectors from the three data sets ALL, F500, and LEV, from which we retrieve cluster-wise probabilities of the noun-verb pairs in the transitive and intransitive cases. Besides, two other algorithms are adopted, which are all described in §4.2. In §4.3, the evaluation methods are explained, followed by technical details in §4.4. The evaluation results and related model selection process is widely described in §4.5. The resulting verb classes are analysed in the light of Levin (1993) classes in §4.6 (qualitatively) and §4.7 (quantitatively).

4.2 Clustering Algorithms

4.2.1 Gaussian Mixture Models: Expectation-Maximization

In machine learning models, we can distinguish two types of clustering: hard-bounded and soft-bounded. Hard clustering entails that clusters do not overlap, i.e. a data point either belongs to a specific cluster or does not, whereas clusters may overlap in soft clustering, i.e. a data point may belong to multiple clusters but with a different degree of belief or weight for each cluster. Mixture models, as opposed to K-Means (see §4.2.3) for example, are a type of soft clustering that is strictly probabilistic. In fact, each cluster is a Gaussian or Multinomial generative model. The distributional parameters of the clusters, ϕ, μ, and σ, are automatically estimated by means of the Expectation-Maximization (EM) algorithm. The EM-based clustering method that I adopt will result in a Gaussian Mixture Model (GMM).

The problem that EM solves is the following: given a data point x_i in a binary classification problem (for the sake of simplicity), not only its source distribution (i.e. class A or class B) is unknown, but also the parameters of this distribution. This fact makes it impossible to guess whether x_i is more likely to belong to a distribution instead of another, since knowledge of ϕ, μ, and σ of both distributions is needed to estimate the source of a set of data points, but at the same time knowledge of the source is needed to estimate ϕ, μ, and σ for class A and class B. The EM algorithm operates in two steps: first, $p(A|x_i)$ and $p(B|x_i)$ are computed so as to determine to what extent it is likely that x_i came from source distribution A (E-step); second, ϕ, μ and σ of A and B are adjusted in order to fit the data points assigned to them (M-step). These steps are iterated until convergence, with the distributions moving from their initialized positions until all examples are included in either one of them. See Dempster, Laird, and Rubin (1977) for further reference.

4.2.2 Spectral Clustering

While an intrinsic evaluation of the Gaussian Mixture Models (see §4.3) is indicator of performance that is relative to specific hyperparameters and data sets, it does not tell how well the algorithm itself does on this task. Specifically, it would be relevant to compare its

performance against other clustering algorithms used in computational semantics, thus to validate how reasonable it is to pursue research in this domain via EM-based modelling. Given the high performance obtained by Sun and Korhonen (2009) on a verb sense classification task, I propose their best machine learning method as term of comparison: Spectral Clustering (SPEC). Although they used a variation based on the MNCut algorithm (Meila & Shi, 2001), I will adopt the more standard clustering method originally proposed by Ng, Jordan, and Weiss (2002), used in Jinxiu Chen, Ji, Tan, and Niu (2006) on an NLP task and in Brew and Schulte im Walde (2002) on German verb clustering and explained later in this section. Moreover, as a baseline, K-means (KM) will be used. As proven in a replication of the experiment of Sun and Korhonen (2009), SPEC is expected to outperform KM also on this task, given that the input data is similar.

In SPEC, clusters are defined by affinity (or adjacency) rather than exact location in the feature space. Concretely, a symmetric affinity matrix G is constructed where $G_{ij} \geq 0$ is a similarity measure (e.g. the quadratic distance) for each data point i with respect to another data point j, given an enumerated sample. By means of Principal Component Analysis (PCA), similar eigenvectors with smallest eigenvalues (except 0, which is distance between a data point and itself) of this rank-deficient matrix are identified and projected to a lower dimensional space[1]. Finally, a clustering method clM is applied on the Laplacian matrix L, i.e. the matrix representation of a graph (since the algorithm is based on graph distance). L is given by:

$$L = D - G, \tag{4.1}$$

where D is the diagonal matrix returned by:

$$D_{ii} = \sum_{j} G_{ij} \tag{4.2}$$

The choice for clM in the last phase of SPEC goes to KM, given that it is our baseline algorithm, making a comparison with SPEC straightforward.

4.2.3 K-Means

KM is one of the most used clustering methods and works as follows. First, a number of centroids are randomly initialized in the feature space. Second, KM will iterate over two steps: 1) data points are assigned to a cluster based on the lowest geometrical distance from the different centroids; 2) the position of the centroids are adjusted by moving them towards the average of the points assigned to each of the components. As already mentioned in Section 4.2.1, one of the differences between EM-based and KM-based clustering is that the former is a form of soft clustering, whereas the latter is an instance of hard clustering. We may expect that, given the complexity of the linguistic data in question, GMMs yield higher performance. In fact, where a hard boundary assigns either a class label or not, a soft boundary allows a data point (e.g. a verb or noun) to belong to different components to different extents in likelihood. Precisely, in terms of verb classes, a verb (or set of verbs) may belong to more than one cluster depending on its sense and on its allowed alternations, causing clusters to be more close to each or even overlap.

[1] In Mathematics, the *spectrum* of a matrix is the set of its eigenvalues, hence Spectral Clustering

4.3 Methods for Cluster Evaluation

The advantage of training a model via unsupervised learning is that little prior assumptions or decisions have to be made about the classes to be drawn. This allows specific patterns in the data to emerge, avoiding biases and confound factors to a large extent that may be caused by the experimental settings. On the other hand, drawbacks are mostly present in the evaluation techniques with regards to a fitted model and in the choice of hyperparameters to adjust with regards to a model to be fitted. Specifically, the data sets that are processed in the present study are unlabeled: we do not dispose of a ground truth value or gold standard that would be necessary to straightforwardly compute a majority class label (purity score) for a specific cluster, and consequently an accuracy measure for the overall model. This fact discards evaluation metrics that require the set of true labels to be known, such as Adjusted Rand index (ARI) used by Mucha and Haimerl (2005) to evaluate hierarchical clustering models in a dialectometry study, and Normalized Mutual Information (NMI) score adopted by Jinying Chen and Palmer (2004) for EM-based cluster analysis in a Chinese verb sense discrimination task. As we will see in §4.3.1, Rooth et al. (1999) compute the accuracy measure through a Pseudo-Disambiguation task. By doing so, we obtain a helpful insight in what the Gaussian components look like, and what the performance differences are among models with a different number of components. At the same time, however, other technical problems arise, as well as a poorly grounded linguistic explanation concerning model selection by Rooth et al. (1999).

I therefore propose alternative evaluation metrics in §4.3.2: Aikake's Information Criterion, Bayesian Information Criterion, and Silhouette Coefficient. Said measures are standardly used for GMM evaluation. Since a single metric is not always able to provide a definitive answer for which model to use, a weighted analysis of these metrics is provided through representation and visual inspection. Furthermore, in §4.3.3, I propose a technique to solve the problem of the number of clusters to be inferred, namely a Variational Bayes Gaussian Mixture Model. The outcomes of the former metrics and the outcome of the latter are then reported and compared in §4.5, which will produce an approximated number of clusters. Finally, in §4.5.4, the latter parameter will be coupled back to the Pseudo-Disambiguation accuracy.

4.3.1 Pseudo-Disambiguation

The clustering models in Rooth et al. (1999) were evaluated on a pseudo-disambiguation (PD) decision task based on Pereira et al. (1993). This method measures the likelihood of a noun n as an argument of a verb v and a verb v', where the pair (v, n) is extracted from the original corpus and the pair (v', n) is artificially composed and completely unseen. The goal of this decision task is to measure how well the model can generalize over unseen verbs, indicating the degree of information that the model has with regards to relations between nouns and verbs.

Concretely, I created an evaluation set E of (v, n, v') triples by cutting a test set E_{test} of 3.000 unique (v, n) pairs out of the original BNC corpus. The original corpus was first transformed into a list $E_{original}$ containing 9.868.709 (v, n) pairs for the ALL data set, 8.847.966 pairs for the F500 data set, and 283.149 pairs for the LEV data set. Consequently, I removed all occurrences of the unique types in E_{test} from $E_{original}$, leaving a reduced training corpus E_{train}. For all three data sets ALL, F500, and LEV, the same transformations and evaluation procedure was applied, but independently the one from another. In this way,

$E_{original}$ and E_{train} had a different size for each data set and related training and evaluation of the resulting models.

Intuitively, v, n, and v' must all three occur in the E_{train}, although in different verb-noun combinations than in E_{test}. Furthermore, verbs and nouns in the E must have a frequency fq in the E_{train} of $30 \leq fq \leq 45.000$. Instances that did not meet this requirement were discarded. The choice for this restriction comes from the consideration that overly frequent verbs and semantically *empty* verbs (i.e. *light verbs)* should not be taken into account in the clustering process, as their contribution is relatively uninformative and could potentially skew the data distribution. Therefore, the upper boundary of 45.000 was chosen, which stands in proportion to the cut-off of 3.000 used by Rooth et al. (1999), given that our total data size is ≈ 15 times larger ($15 \times 3.000 = 45.000$). The lower boundary of 30 indicates the necessity of a minimum number of examples in order to render the vectors sufficiently robust.

This downsizing from the initial evaluation sample of 3.000 $v, n)$ pairs, due to these restrictions, brings us to a final set of evaluation triples E of 2.404 (v, n, v') types for the ALL data set, 1.932 for the F500 data set, and 2.498 for the LEV data set.

The PD accuracy is calculated by counting how many times the latent class model $p_{LC}(\cdot)$ returns a probability that is higher for verb-noun pairs (v, n) (that do exist in the original corpus) than for verb-noun pairs (v', n) (that do *not* exist in the original corpus), out of all evaluation triples (v, n, v') in E. The triples for which this relationship holds, are members of E', a subset of E. Hence, the PD accuracy is given by the cardinality of E' in proportion to its superset E. Formally:

$$E' = \{(v, n, v') \in E \mid p_{LC}(n|v) \geq p_{LC}(n|v')\} \tag{4.3}$$

$$PD_{accuracy} = \frac{|E'|}{|E|} \tag{4.4}$$

4.3.2 Density, Separateness and Complexity of the Clusters

Since the model trained on the entire BNC instead of fitting a sole subpart of it as was done in Rooth et al. (1999), we can question what the measure of generalization power actually tells us. If we consider the BNC as a representative reflection of the English language, than the model would not need to perform decently on unseen data, as ulterior (v, n) occurrences would be either contained in the training data, or be relatively rare and thus, arguably irrelevant. In that case, the Pseudo-Disambiguation task only gives us an intuition about the model's performance, but no attached applicability value. In order to evaluate the models from a different perspective, we could inspect more closely the clusters that are produced by more informative metrics, such as (i) Silhouette Coefficient (SC), (ii) Akaike's Information Criterion (AIC) and (iii) Bayesian Information Criterion (BIC). In the process, also the Log-Likelihood Value (LLV) of the models will be taken into account. First, SC indicates how the clusters that are produced by the model with respect to a sample are formed. Specifically, it tells how well separated the clusters are and it provides an indication of the distance of the data points from their respective centroids. The SC for a model is computed as

$$SC = \frac{(n - i)}{max(i, n)}, \tag{4.5}$$

where n is the mean nearest-cluster distance of all data points from the cluster into which they have not been classified, and where i is the mean intra-cluster distance. Results deriving from this metric range from -1 to 1: if the SC tends to -1, samples are misclassified to a larger extent than the case in which the SC tends to 1; if the SC approaches 0, clusters do overlap.

Second and third, it is worth analyzing the fitted models by means of the AIC and BIC, which are two likelihood criteria that set a penalty based on the complexity of the model. Formally, the BIC is given by

$$BIC = -2\ln(L) + k\ln(n), \tag{4.6}$$

with L being the maximum likelihood value of the model, k the number of parameters, and n the number of observations. Similarly, the AIC is given by

$$AIC = 2k - 2\ln(L) \tag{4.7}$$

The difference between these two types of information criteria can be explained by the different penalty weight that is applied for the number of parameters: $2k$ and $\ln(n)k$ for AIC and BIC, respectively. Thus, we can conclude that the BIC will penalize the model more heavily than the AIC. For both metrics, the lower the score, the better the model.

4.3.3 Variational Bayesian Estimation of a Gaussian Mixture

Since we are aiming at a fully unsupervised method to avoid as much prior bias as possible on the number of clusters to be created, I propose to adopt a Bayesian variant of a Gaussian Mixture Model. Whereas EM yields a probability distribution over the clusters together with a maximum likelihood estimate θ, its Bayesian extension draws a probability distribution over θ and the latent variables. Through Variational Bayes (VB) estimation, θ is treated as an extra latent variable, optimizing each of them one at a time. Hence, VB computes estimates of the posterior distribution of all variables, both parameters and latent variables, after first fitting a prior distribution to these parameters. Based on the weights of this prior distribution, a VB Gaussian Mixture can infer its true number of components automatically from the data, by setting posterior weights of the components to probabilities that are close to zero. In this way, the model will highlight the components that best fit the data, ignoring irrelevant ones. A distinction between types of components based on their active role in the model will be explained later on in this section.

In detail, two types of prior mechanisms can be used: Dirichlet process prior and Dirichlet distribution prior. The former defines an infinite number of components and activates only those that are necessary. The latter defines a finite mixture model with a finite number of components, favoring a more uniform weight distribution over the components. This difference leads to more natural classes in the first case, whereas the second option tends to divide otherwise natural classes into sub-components that are active themselves (as each sub-component would gain relatively more weight than the Dirichlet process prior would allow). In view of these differences, we consider the Dirichlet process prior as the most appropriate option to apply as less influence as possible on the natural patterns in our linguistic data.

The main hyperparameter to be tuned is the number of components. Unlike for regular Gaussian Mixture Models, this hyperparameter is not the effective number of clusters to be formed, but an upper-bound. This value is the maximum number of clusters that is *a*

priori believed to cover the true generative distribution of this data set. Furthermore, the upper-bound turns out to be a summation of the redundant (inactive) and effective (active) components in the fitted model, given the posterior weights for each component. Redundant components will be identified by a weight that is close to 0 as produced by the algorithm. Hence, the task is to identify a lower cut-off for the posterior weights distribution, so as to establish which components are to be regarded either as active or inactive. The posterior weights distribution is closely connected to a *weight_prior* parameter, which indicates the bayesian prior distribution that is believed to correctly describe the natural clusters in the data. This parameter can be adjusted to a high value if it is expected that the data in question is structured in many clusters, by giving more weight to smaller groups of data points. The same parameter can be tuned to a lower value if it is desirable to give more weight to *the big picture* in the data, hence, a smaller number of components. Our assumption is that the prior weights are equally distributed over the components, indicating a parameter choice of $weight\_prior = \frac{1}{n\_components}$. The motivation for this is that we do not want to influence the result of the VB GMM by the results of the metrics proposed in §4.3.2 that could favor either a relatively low or relatively high number of components for the different GMMs. For further reference on Variational Bayes Gaussian Mixture Models consult Blei, Ng, and Jordan (2003)).

4.4 Fitting the Models: Implementation Details

The models were trained with the Scikit-learn package (Pedregosa et al., 2011), version 0.20.3. With exception for the Variational Bayes Gaussian Mixture Model — which had a runtime of between 2 and 3 hours for the ALL data set — fitting the models lasted between 20 and 30 minutes for the ALL data set, less then 15 minutes for the F500 data set, and less than 1 minute for the LEV data set. For visualization, Matplotlib (Hunter, 2007), version 3.0.3 was used.

4.5 Model Selection

The selection of the *best* model depends on several factors, such as (i) the number of clusters — to which little importance had been given in Rooth et al. (1999) —, (ii) the choice between the entire BNC data set (ALL) and its different subparts (F500 and LEV), and (iii) the types of metrics (e.g. accuracy, intra-cluster distance, Variational Bayesian inference). I will begin with showing the results of the evaluation method in Rooth et al. (1999): Pseudo-Disambiguation (PD).

4.5.1 Pseudo-Disambiguation (Results)

As we may recall, this measure returns an accuracy score (ACC) indicating the extent to which the model is able to generalize. In Figure 4.1, the variation of ACC is displayed in relation to the number of clusters ranging from 5 to 100, to the given data set, and to the clustering algorithm that was used. For all three data sets, the GMM accuracy has an approximate positive correlation with the number of clusters, in ranges of around $60\% - 70\%$ accuracy for ALL and F500, and $70\% - 83\%$ for LEV. Nevertheless, the result of the PD task will be taken into account in a different manner than in Rooth et al. (1999). In contrast to the latter, I do not limit the analysis to the ACC curve over the number of components as the representation of a general trend. Instead: (1) I compare the ACC curves for GMM,

SPEC, and KM so to obtain an intuition regarding the measure to which GMM was a good algorithm choice overall for this NLP task; (2) only after computing the optimal number of components (per data set) via an in-depth cluster analysis, I retrieve the ACC value that is related to that number. For example, if the best number of clusters appears to be 50 according to the in-depth cluster analysis, we calculate the PD accuracy for a model with 50 clusters. Even though we will come back to the PD accuracy later, I first give an overview of the overall PD performance for both the different data sets and the different algorithms.

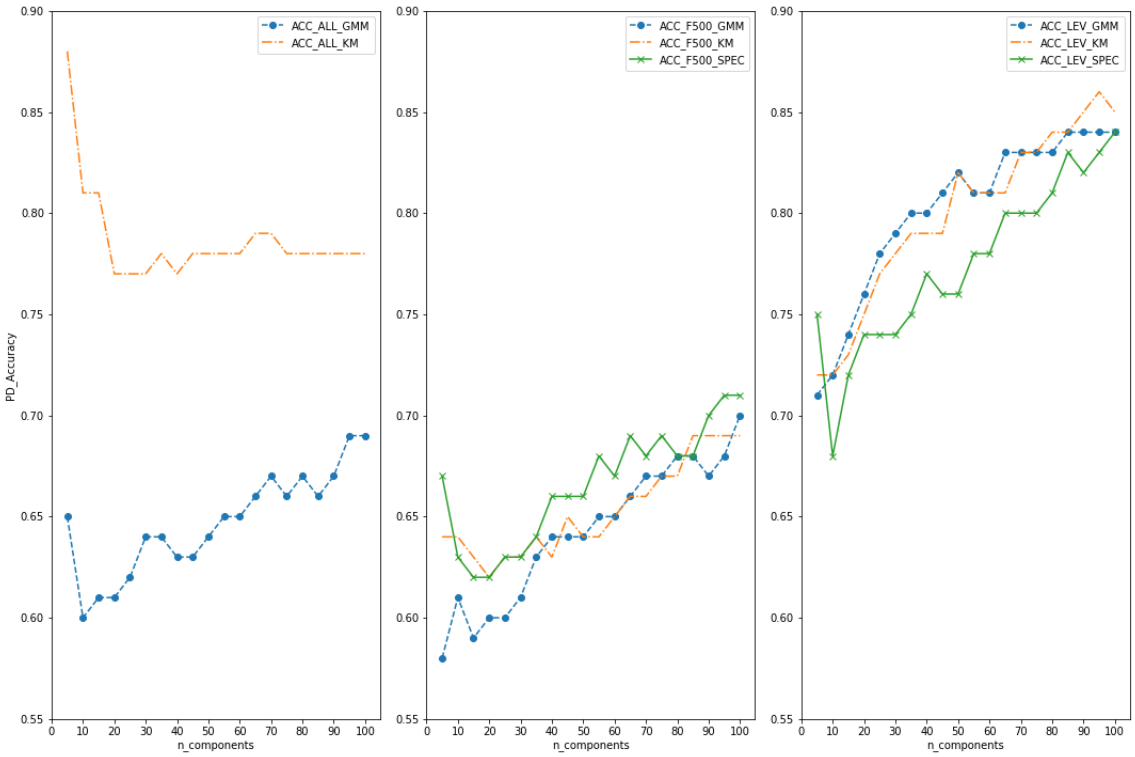

Figure 4.1: Pseudo-Disambiguation Accuracy comparison between all verbs (ALL_ACC), the 500 most frequent verbs (F500_ACC), and Levin's transitive-intransitive alternation verbs (LEV_ACC). Three different clustering techniques are used in combination with these data sets: Gaussian Mixture Models (GMM), K-means (KM), and Spectral Clustering (SPEC).

According to Figure 4.1, the difference in PD performance between GMM and SPEC is minimal but present for the two data sets on which it was applied, with SPEC scoring relatively higher than GMM on F500, but lower on LEV. This fact is relatively surprising given that LEV is smaller in size than F500, and SPEC is known to perform well on reduced data sets. In fact, its applicability to large-scale problems is restricted because of its computational complexity of $O(t^3)$, where t represents the number of data points (Yan, Huang, & Jordan, 2009). On the other hand, KM achieves a similar performance on F500 and LEV in comparison to GMM, but scores clearly higher on the ALL data set: $> 75\%$ versus $60\% - 70\%$. We can infer from this that KM performs sensibly better on highly sparse data sets in comparison to EM. Concerning (2), the in-depth analysis consists of the techniques described in §4.3.2 and §4.3.3: on the one side the standard metrics AIC, BIC, SC, and LVV, while on the other we find an inspection of the posterior weight distribution

of a fitted VB Gaussian Mixture Model on the same three data sets.

4.5.2 AIC, BIC, SC, and LLV (Results)

The GMM in-depth cluster evaluation results can be observed in Figure 4.2 (ALL data set), Figure 4.3 (F500 data set), and Figure 4.4 (LEV data set).

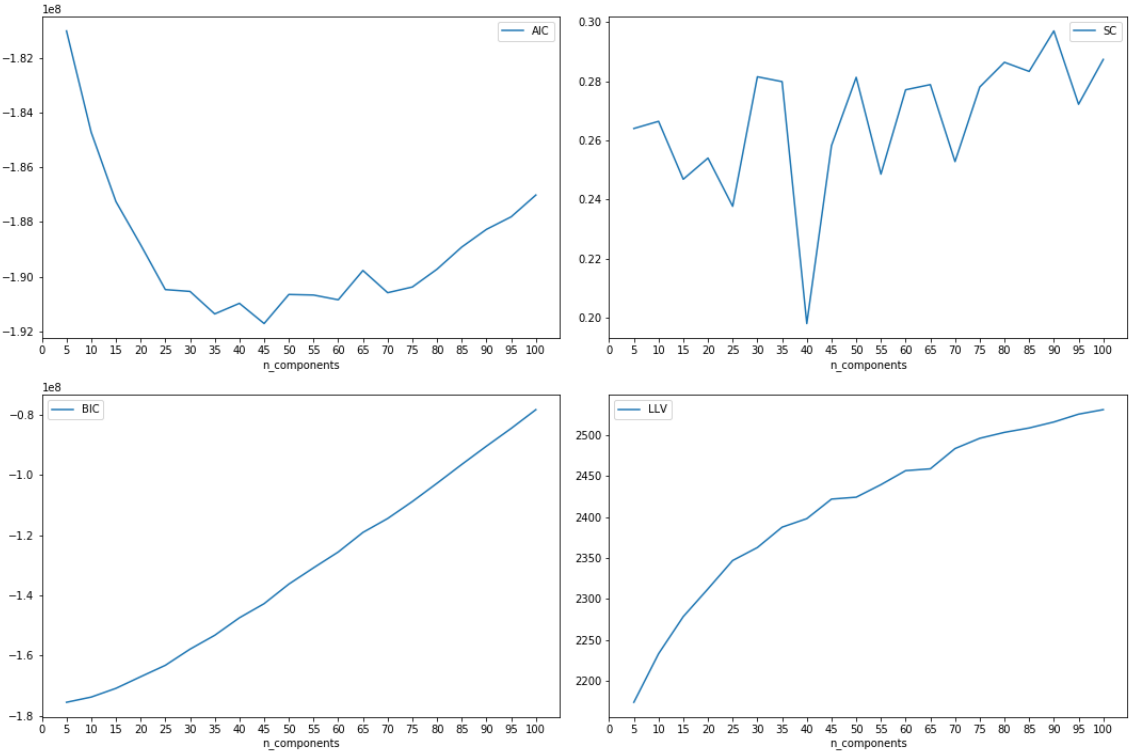

Figure 4.2: GMM evaluation on ALL data set. Metrics: AIC, BIC, SC, LLV

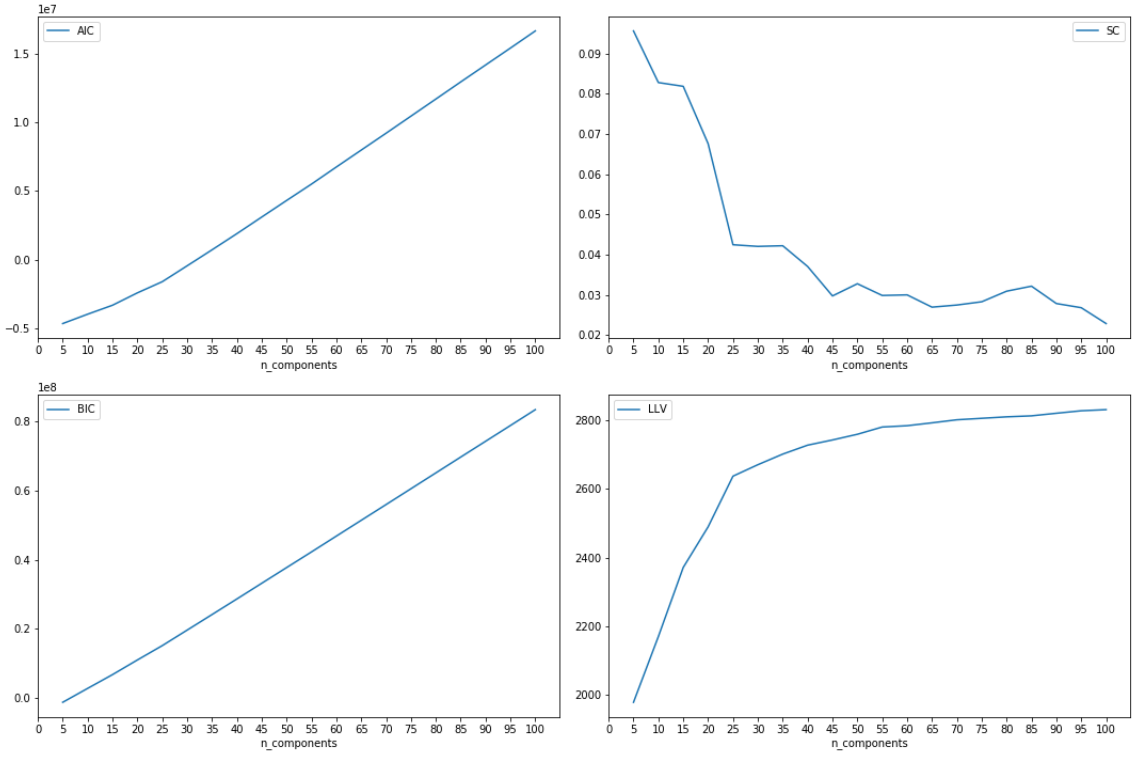

Figure 4.3: GMM evaluation on F500 data set. Metrics: AIC, BIC, SC, LLV

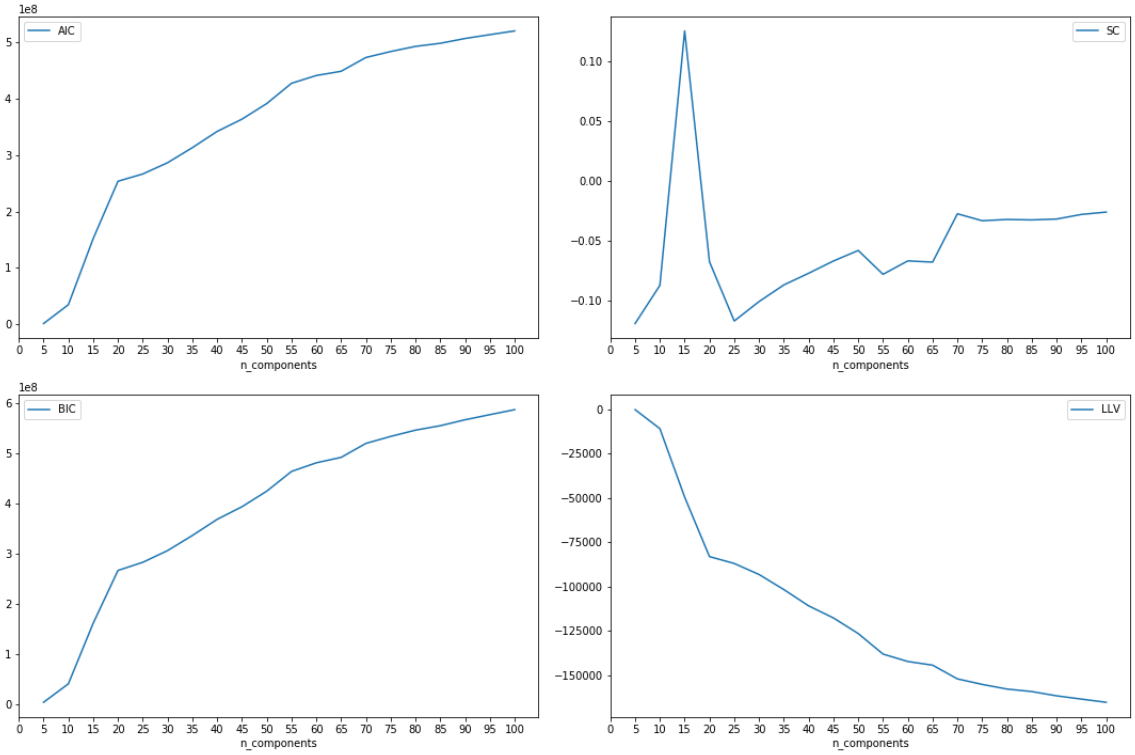

Figure 4.4: GMM evaluation on LEV data set. Metrics: AIC, BIC, SC, LLV

The AIC and BIC are penalty-based metrics that directly depend on the complexity of the model, which in our case is the number of components. The way in which Rooth et al. (1999) interpreted the models' performance, i.e. by the sole PD accuracy related to a range of models in a window of $[25; 100]$ components, does not deal with this aspect. Furthermore, they showed the results of the semantic slot labeling experiment based on a model with 35 clusters, without accounting for that specific parameter.

For the ALL data set (Figure 4.2), although a higher number of components is more likely to explain the data (LLV: *the higher, the better*), the AIC and BIC prefer a low number of components in the range $[5; 100]$ that is given (*the lower, the better*). The AIC is particularly informative as it rapidly decreases between 25 and 75 components, with a negative peak around 35 and 45. For this data set, but also for F500 and LEV as we will observe, BIC is not the most appropriate metric (relatively uninformative lines or curves). As expected for a soft-clustering algorithm like EM, the SC oscillates around 0. Besides, its values are positive, indicating that data points are not assigned to the wrong cluster. High peaks are registered starting from the model with 30 components and is consistent with the intuition given by the AIC: given the peak at 35 components (despite being lower at 45), this parameter option is confirmed as a valid candidate. A (local) maximum is registered at 90 components, but due to the BIC we give preference to a lower model complexity.

For the F500 data set, by observing Figure 4.3 it appears that the AIC is penalizing the model's complexity in a similar way to the BIC, without showing a reverse bell-shape (and consequently a reduced preference window) as for the ALL data set. A higher precision rate comes with SC and LVV, which indicate a preference for a window between 15 and 35 as a valid trade-off.

Ultimately, for the LEV data set in Figure 4.4, we can clearly observe that the optimal value for the number of components is 15. Specifically, both the AIC and BIC favor a number of components under 20, with the LVV showing a related decrease at the same value. At the same time, the SC points to 15 as the best value for the number of components being not only the highest, but also the only positive SC among the range of *n_components* candidates.

In the next section, §4.3.3, an extra evaluation metric is given for the estimation of the number of components per model. Its outcome, together with an overview of the results in the present section, is given in §4.5.4.

4.5.3 Variational Bayes (Results)

In order to render the result of the cluster evaluation more robust, and thereby obtain a strong estimate of the number of lexical verb categories in the transitive-intransitive alternation for each data set, we apply a technique that is unseen in this domain so far. Until now, we empirically compared models that were differing in the choice of the main hyperparameter: the number of components. However, as explained in §4.3.3, this value can be estimated automatically from the input data by means of Variational Bayes (VB) inference. By analysing the weight distribution over the fitted clusters in a VB GMM, the effective number of components is equal to the established upper-bound (50, in our case) minus the number of redundant components. As we may recall, redundant or inactive components are those that have a weight that is driven towards 0. Figure 4.5 shows the results of three VB GMMs, one for each data set that we used so far. In particular, the histograms illustrate the portion of components that has a posterior weight close to 0,

indicated in red[2]. Concretely, we may consider as close to 0 the bin of weights in each histogram with lowest value on the x-axis, which has the characteristic of being the highest on the y-axis since VB GMMs tend to give a strong bias towards 0 to a relatively high portion of upper-bound number of components. Hence, from this analysis we can deduce that the number of redundant components in Figure 4.5 with respect to the different data sets, is 24 for ALL, 17 for F500, and 30 for LEV. In other words, the VB inference technique points to 26, 33, and 20, respectively, as approximations for the optimal number of clusters.

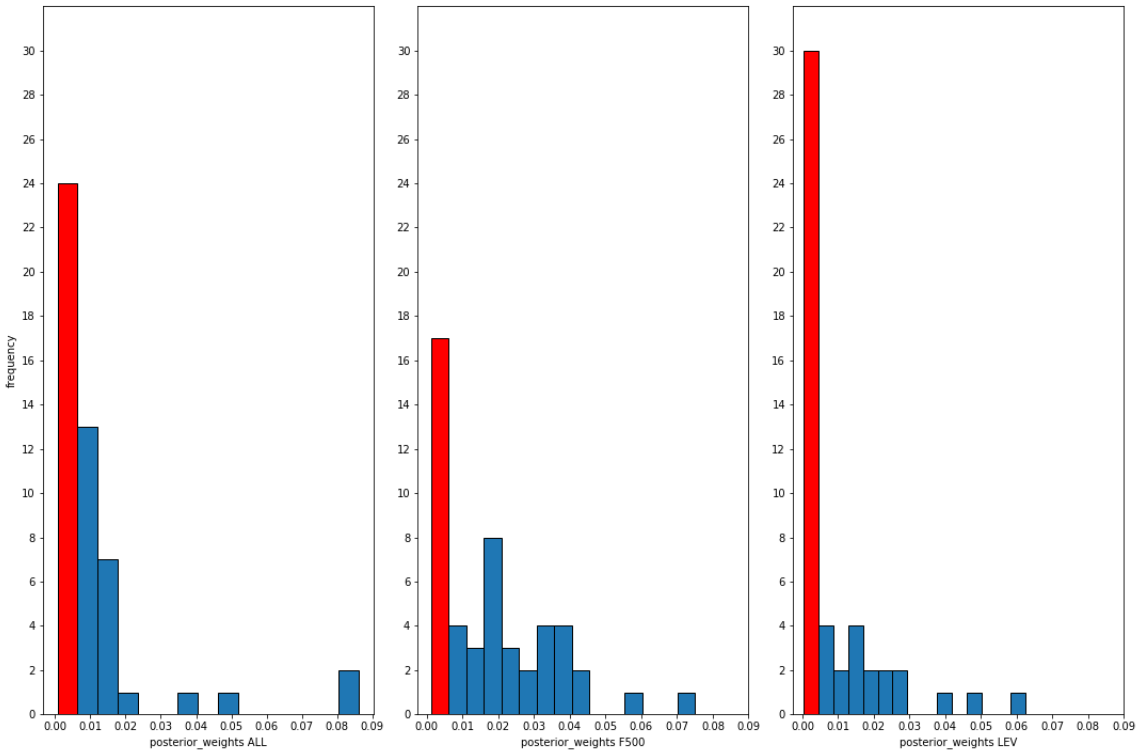

Figure 4.5: The frequency of redundant/inactive components (marked in red) versus effective/active components (marked in blue) in the VB GMM model for the three data sets ALL, F500, and LEV, with an initial upper-bound of 50 components.

4.5.4 Putting Everything Together

We may now join together all metrics that were computed in the present chapter in order to come up with a balanced decision on the number of components for each data set. Table 4.1 provides an overview of the predictions of said metrics. By taking both types of evaluation scores into account, I propose that combination of AIC, BIC, SC, and LVV is used as a baseline approximation, which is then refined by the result produced through VB inference.

Concretely, we saw that the ALL-model has a preference for 35 or 45 clusters given by the first series of metrics; out of these two candidates we may decide for the lowest, being it a fair in between choice with respect to the value of 26 given by VB.

[2]Outlier posterior weights with a value of > 0.01 (relative to a single component in each model) were ignored in the plot so to represent on a more informative scale the difference in frequency between component weights close to 0.

Active n_components	*Metric*	
Data set	*AIC, BIC, SC, LVV*	*VB*
ALL	35/45	26
F500	15-35	33
LEV	15	20

Table 4.1: Optimal number of active components in a GMM given the type of metric and the data set.

As for the F500-model, the uncertainty of the options window $15-35$ is strongly reduced by the result of VB: a higher number of clusters is preferred, therefore a value of 35 would be a well-founded choice.

Finally, for the LEV-model the values returned by the two types of metric are very close to each other, but a preference is given to 15 rather than to 20 because of the clear peak given by the SC metric in §4.5.2.

In Table 4.2, I report the final choices about the number of components and, consequently, the model(s) to be used as a starting point for the semantic role annotation phase. Here, based on the intrinsic cluster evaluation widely explained in the present section, we refer back to the Pseudo-Disambiguation results that were shown in Figure 4.1.

Data set	*n_components*	*PD-accuracy*
ALL	35	63.98%
F500	35	63.37%
LEV	15	74.1%

Table 4.2: Final n_components parameter for each GMM and relative Pseudo-Disambiguation Accuracy.

From Table 4.2 it clearly appears that the F500 data set is representative of its superset, i.e. the ALL data set. In fact, the lexical structures of both data sets are likely to be divisible into 35 components, yielding a similar performance in terms of accuracy ($63.4\% - 64.0\%$). Between the two models, then, usually the simplest is chosen, namely the F500-based. However, KM-based clustering yields sensibly higher results on the ALL data set, but fails to outperform GMM on the F500 data set. What we can infer from this, is that F500-based models fail to generalize for the low-frequence occurrences in the data. Moreover, it is possible that the LEV data set is structured more precisely than F500 and ALL (i.e. composed of true alternating verbs according to Levin, 1993), as indicated by an accuracy score of 74.1%.

Compared to the performance obtained in previous work (as introduced in §2.2.3 and §2.2.4), the GMM accuracies for the ALL and F500 data set obtained in the present study are not in line with Rooth et al. (1999), but they are similar to Erk (2007)'s model's performance. The use of K-means, instead of GMM, could possibly equal the score of 80% on said data sets. On the other hand, the verbs in LEV contributed to a comparable score with respect to Rooth et al. (1999).

4.6 Results and Qualitative Analysis

The concrete outcomes of the EM clustering procedure are briefly treated in this section, in the form of a visual inspection and qualitative analysis. A quantitative analysis follows in §4.7.

In the light of the reasons provided in §4.5, I consider the model that is based on the LEV data set because of its highest accuracy, and the GMM based on the ALL data set because of its wider coverage. The F500-model, which (as we may recall) covers approximately 90% of the data fitted by the ALL-model, does not score better than the latter, and is outperformed by the KM algorithm. Then, let us inspect the clusters to gain a better insight — in both the lexical and statistical sense — in the conditional probabilities of verbal and nominal instances that are grouped together by the models.

Starting with the LEV-model with 15 components, two arbitrary examples of automatically induced verb classes are given in Table 4.3 and Tables 4.4. The two overviews list the most probable verbs given a cluster with an arbitrary label 7 and 5, as well as the most probable nouns conditioned on the same cluster.

Class 7 (LEV)			
$p(v\|7)$		$p(n\|7)$	
alter.tr:o	0.3383	matron	0.3841
increase.intr:s	0.3265	fiend	0.3529
expand.intr:s	0.057	lattre	0.2758
loosen.tr:s	0.0549	canister	0.2275
decrease.intr:s	0.0364	opening	0.1471
float.intr:s	0.0361	lay-out	0.081
compress.tr:s	0.0279	kloppenberg	0.0774
compress.tr:o	0.0276	smile	0.0754
reopen.tr:s	0.0261	screen	0.0681
straighten.intr:s	0.0218	aws	0.0681
squirt.tr:s	0.0124	replacement	0.0676
solidify.tr:s	0.0109	brushwork	0.0638
spin.intr:s	0.0103	president	0.0591
energize.tr:s	0.0082	making	0.0547
lighten.intr:s	0.0027	discontent	0.0481

Table 4.3: Class 7 of the GMM with 15 components based on the LEV data set.

Class 5 (LEV)			
$p(v\|5)$		$p(n\|5)$	
open.tr:o	0.2945	wielder	0.5756
move.intr:s	0.1181	basildon	0.3493
stand.tr:s	0.1072	melms	0.1653
walk.intr:s	0.0997	phi	0.1566
close.tr:o	0.0973	wolf	0.1141
improve.tr:s	0.0396	bunny	0.1107
sit.tr:o	0.0346	raid	0.0982
vary.intr:s	0.0332	plethora	0.0753
double.tr:o	0.0166	tutor	0.0595
swing.tr:s	0.0149	hussey	0.059
fold.tr:s	0.0134	jaq	0.0542
flash.tr:s	0.0105	massacre	0.0532
bang.tr:s	0.0094	know-how	0.0532
bang.tr:o	0.009	yeltsin	0.0489
lodge.tr:s	0.0084	marseillaise	0.0474

Table 4.4: Class 5 of the GMM with 15 components based on the LEV data set.

Tables 4.5 and 4.6, on the other hand, illustrate the most probable nouns and verbs in the clusters 10 and 16 of the 35-component model based on the ALL data set. At a first glance, we could observe and recognize certain lexical patterns (i.e. consistent semantic classes) in these sets, and perhaps acknowledge a subtle difference between the LEV clusters and the ALL clusters reflecting the difference in accuracy of the two models. However, such observations should be backed up with quantitative arguments, which is a part that is missing in linguistic interpretation by Rooth et al. (1999), but provided in the next section.

Class 10 (ALL)			
$p(v\|10)$		$p(n\|10)$	
get.tr:s	0.2149	adulterer	0.1683
want.tr:s	0.1323	crushing	0.1399
require.tr:s	0.1014	tammuz	0.1233
win.tr:o	0.089	adjudge	0.1225
like.tr:o	0.069	xyz	0.1208
sell.tr:o	0.0329	rodent	0.1075
achieve.tr:s	0.0264	orchids	0.1029
cross.tr:o	0.0245	ones	0.098
eat.tr:s	0.0223	scriptures	0.0917
stop.tr:s	0.0165	gnat	0.0899
define.tr:o	0.0152	respirator	0.0831
live.tr:s	0.0143	dependant	0.0806
oppose.tr:s	0.0128	graptolites	0.0772
fight.tr:s	0.0115	archdeacon	0.0749
deliver.tr:o	0.011	barricade	0.072

Table 4.5: Class 10 of the GMM with 35 components based on the ALL data set.

Class 16 (ALL)			
$p(v\|16)$		$p(n\|16)$	
ask.tr:s	0.1583	partnership	0.1661
claim.tr:o	0.0783	prognosis	0.1463
know.intr:s	0.0783	illness	0.1439
mention.tr:s	0.0733	potentiation	0.1328
encounter.tr:s	0.0483	incidence	0.1268
note.tr:o	0.0425	hartford	0.1189
improve.tr:o	0.0341	morbidity	0.1142
ensure.tr:o	0.0334	coincidence	0.1075
unveil.tr:s	0.0264	sheldrake	0.1007
restrict.tr:s	0.0256	familiarity	0.098
restore.tr:o	0.0227	scooper	0.0961
hide.tr:s	0.0222	functioning	0.0842
wave.tr:o	0.0219	smooth	0.0777
stage.tr:s	0.0207	dorothy	0.0733
satisfy.tr:o	0.0164	ringwraith	0.0718

Table 4.6: Class 16 of the GMM with 35 components based on the ALL data set.

4.7 A Quantitative Analysis of the Models' Verb Classes

The models have been extensively evaluated in §4.5 and the results described in §4.6. Data-wise, the evaluation is well-grounded, but a qualitative analysis of the verb classes may be insufficient for a holistic impression of the results. More precisely: how semantically coherent are the verb classes produced by the models? Whereas Rooth et al. (1999) do not provide a quantitative analysis, I propose a semantic evaluation method based on Sun and Korhonen (2009) that maps back to Levin (1993)'s verb classes. Sun and Korhonen (2009) evaluated the results of their unsupervised clustering experiment, which I described in §2.2.2.4, by means of two test sets T1 and T2 that were originally used by Sun, Korhonen, and Krymolowski (2008) and Joanis et al. (2008), respectively. Both T1 and T2 are subsets of verbs retrieved from 15 to 17 Levin (1993) classes. The classes in T1 were selected at random with one of the constraints being that each class had enough member verbs whose predominant sense belongs to the class in question. In general, our task differs from Sun and Korhonen (2009) in the sense that I did not create a data set of verbs from particular Levin (1993) classes (although the LEV data set is based on a particular *alternation*, but not a *class*), nor are the verb vectors of the same type (one vector = one verbal entry in their approach, versus three vectors = one verbal entry in our approach). Specifically, Sun and Korhonen (2009) first decided on a set of Levin classes to test the clustering results against, with every class covering between 10 and 20 different verbs. Key in their approach was the fact that every verb was given a specific label of the class to which it belonged, besides the property of being mostly monosemous.

Since I started with an unlabeled set of verbs instead, I adopt an inverse procedure by labeling the verbs with one or more Levin classes *a posteriori*. Only intransitive verbs are taken into account, as transitive verbs are divided into two vectors and may therefore be divided over two different clusters. First, through VerbNet (Kipper-Schuler, 2005), a list of one or multiple coarse-grained Levin class labels (out of a total of 274) was assigned to each verb. To perform this labeling procedure, the NLTK (Loper & Bird, 2002) the

VerbNet corpus reader was used. Second, the models were evaluated by means of cluster-wise measurements.

The evaluation metrics were adopted not only by Sun and Korhonen (2009), but also by other studies in this field such as Korhonen, Krymolowski, and Collier (2008) and Ó Séaghdha and Copestake (2008). Two metrics were used: a modified purity measure $mPUR$ and a weighted class accuracy $wACC$; the two can be regarded as a precision and recall measure, respectively, which are required to compute a weighted average F-measure. $mPUR$ evaluates the mean precision of clusters, which are labelled according to their prevalent class. $wACC$ is the proportion of members of dominant clusters within all classes. The three metrics $mPUR$, $wACC$, and F-measure are defined as follows:

$$mPUR = \frac{\Sigma_{n_{prevalent(k_i)>2}} n_{prevalent(k_i)}}{n\_verbs} \tag{4.8}$$

$$ACC = \frac{\Sigma_{i=1}^{C} verbsinDOM\_CLUST_i}{n\_verbs} \tag{4.9}$$

$$F = \frac{2 \cdot mPUR \cdot ACC}{mPUR + ACC} \tag{4.10}$$

The random baseline BL is computed as follows:

$$BL = \frac{1}{n\_components} \tag{4.11}$$

This quantitative evaluation of the induced verb classes as in relation to those proposed by Levin (1993) was applied on the LEV and ALL data set models. Their respective outcomes can be observed in Table 4.7.

Data Set	*n_components*	*mPUR*	*wACC*	*F*	*BL*
ALL	35	24.56%	36.32%	29.30%	2.86%
LEV	15	24.52%	32.09%	27.80%	6.67%

Table 4.7: Results of the Quantitative Analysis with respect to the verb classes that were produced by the ALL-based model and the LEV-based model.

The first thing to notice here, is that both F-measures outperform the baseline: 29.30% versus 2.86% and 27.80% versus 6.67%. The second observation is that the scores are similar for the two data sets. Compared to the F-scores in Sun and Korhonen (2009) however — which reached values of 80% for their task — the performance is certainly lower. Intuitively, this may be an expected outcome. In fact, our task does not include the goal of necessarily recreating Levin (1993)-alike verb classes, and the results are affected to an important extent by the transitive feature vectors, which also contribute information to the fitting of the clusters. With respect to Rooth et al. (1999) and our qualitative analysis in §4.6, a quantitative semantic analysis returns statistical information that concerns all 15 clusters for the LEV-based model and all 35 clusters fro the ALL-based model, as well as the totality of the verbs that are member of these clusters. On the contrary, the qualitative analysis certainly provides an intuitive idea of how the most probable verbs for some clusters are grouped together, but it is also true that verbs with low absolute frequencies may not be clustered in a coherently semantic fashion.

CHAPTER 5
EXPERIMENT 2

5.1 Task: Semantic Role Labeling with Latent Classes

The second part of this study consists of a re-estimation of the probabilities returned by our best Gaussian Mixture Model, performed by a further EM-based inference step. Differently from the first part of the experiment, where we obtained the probabilities of the nouns and verbs per cluster, here we compute estimated frequencies of the argument nouns of fixed verbs, in order to induce verb-specific lexicons for its argument slots. The formalization that follows below of this procedure is directly taken from Rooth et al. (1999).

Given a latent class model $p_{LC}(.)$ for verb-noun pairs, and a sample $n_1, ..., n_M$ of subjects for a fixed intransitive verb, we calculate the probability of an arbitrary subject $n \epsilon N$ by:

$$p(n) = \Sigma_{c \epsilon C} p(c, n) = \Sigma_{c \epsilon C} p(c) p_{LC}(n|c) \tag{5.1}$$

The estimation of the parameter-vector $\theta =< \theta | c \epsilon C >$ can be formalized in the EM framework by viewing $p(n)$ or $p(c, n)$ as a function of θ for fixed $p_{LC}(\cdot)$. The re-estimation formulae resulting from the incomplete data estimation for these probability functions have the following form ($f(n)$ is the frequency of n in the sample of subjects of the fixed verb):

$$M(\theta_c = \frac{\Sigma_{n \epsilon N} f(n) p_\theta(c|n)}{\Sigma_{n \epsilon N} f(n)} \tag{5.2}$$

Furthermore, for verbs in the transitive form, the estimation of the two filler nouns n_1, n_2 and related clusters c_1, c_2 is required. We induce latent semantic annotations for transitive verb frames. Given a LC model $p_{LC}(\cdot)$ for verb-noun pairs, and a sample $(n_1, n_2)_1, ..., (n_1, n_2)_M$ of noun arguments (n_1 subjects, and n_2 objects) for a fixed transitive verb, we calculate the probability of its noun argument pairs by:

$$p(n_1, n_2) = \Sigma_{c_1, c_1 \epsilon C} p(c_1, c_2, n_1, n_2) = \Sigma_{c_1, c_1 \epsilon C} p(c_1, c_1) p_{LC}(n_1|c_1) p_{LC}(n_2|c_2) \tag{5.3}$$

Again, estimation of the parameter-vector $\theta =< \theta_{c_1, c_2} | c_1, c_2 \epsilon C >$ can be formalized in an EM framework by viewing $p(n_1, n_2)$ or $p(c_1, c_2, n_1, n_2)$ as a function of *theta* for fixed $p_{LC}(\cdot)$. The re-estimation formulae resulting from this incomplete data estimation problem have the following simple form ($f(n_1, n_2)$ is the frequency of (n_1, n_2) in the sample of noun argument pairs of the fixed verb):

$$M(\theta_{c_1, c_2} = \frac{\Sigma_{n_1, n_2 \epsilon N} f(n_1, n_2) p_\theta(c_1, c_2|n_1, n_2)}{\Sigma_{n_1, n_2 \epsilon N} f(n_1, n_2)} \tag{5.4}$$

The resulting re-estimations were not computed with any packages, but *manually* extracted from the fitted models. In fact, each cluster displays a series of (conditional) probabilities given their composition of verb vectorizations, which make it straightforward to recompute the frequencies $f(n) p_\theta(c|n)$ for the nouns given a verb.

5.2 Results and Qualitative Analysis

The slot labeling experiment, introduced in §5.1, produced a series of examples of transitive and intransitive verbs along with their re-estimated frequencies. A sample of them is shown in this section.

Table 5.1 shows six intransitive verbs with their respective subject head fillers. The choice of showing precisely these verbs is not arbitrary: *blush*, *snarl*, and *increase* are also reported as examples by Rooth et al. (1999) and can serve as qualitative benchmark with respect to the original study; apart from *increase*, also *break*, *melt*, and *slow* are included in the LEV set of verbs, hence they will be used in a qualitative comparison between the ALL data set and the LEV data set. It must be noticed that the verbs that are present in the LEV data set are overall infrequent. At first sight, we may observe a decent overlap between the estimated lexicon for *blush* in Table 5.1 and for *blush* by Rooth et al. (1999), both consisting of predominantly female proper names, among which even of a couple of exact the same names. On the other hand, the lexicon produced for *snarl* does overlap less. The other four verbs look reasonable in the light of the model's accuracy of $\approx 65\%$.

Intransitive Verb Subject Slot Re-estimations (ALL)					
blush		*snarl*		*increase*	
she	1.6758	masklike	1.0	turnover	2.1802
anabelle	0.3535	clueless	0.949	cost	1.0038
lou	0.3025	he	0.2747	dalles	1.0
sarah	0.2103	she	0.2095	subscription	1.0
he	0.1479	alsatian	0.1581	note-rate	1.0
cottle	0.1477	man	0.1084	inmigration	1.0
maggie	0.1123	spider	0.0639	ornithischosus	1.0
willie	0.0963	craon	0.061	crop-raiding	1.0
constance	0.0791	sabrina	0.0592	agrarianism	0.9033
year	0.0735	president	0.0438	income	0.8916
break		*melt*		*slow*	
fire	2.05	heart	1.2495	newman	1.4911
light	1.3554	snow-wreaths	1.0	punk-neutrons	1.0
she	1.065	englishwomen	1.0	goalscoring	1.0
window	1.0434	ambitions	0.7486	frisbee	0.9345
pen-nib	1.0	gelatine	0.4783	housebuilding	0.6876
scuffle	1.0	rainbows	0.4458	depreciation	0.6026
kneecap	1.0	soldiery	0.3686	collagen	0.5899
police	0.7586	sinew	0.3159	sickle	0.5473
wave	0.5813	snow	0.228	expansion	0.5364
war	0.5732	teacher	0.2074	growth	0.5244

Table 5.1: Re-estimations for six intransitive verbs from the ALL data set model, in descending order of $f(n)p_\theta(c|n)$

.

In Table 5.2, again six verbs are shown, this time for the LEV-based model. Since *blush* and *snarl* make not part out of the LEV set of verbs, I replaced them with two arbitrary examples *fly* and *harden*. By analysing Table 5.1 against Table 5.2 in the form of pairwise comparisons between the induced lexicons of the same verb, but with different source, we

can notice that the re-estimation frequencies are overall higher for the LEV-based model. This derives from the fact that, compared to ALL, the LEV data set is smaller in size, and less sparse as a consequence.

Intransitive Verb Subject Slot Re-estimations (LEV)					
fly		*harden*		*increase*	
spark	3.5674	face	1.3035	rate	3.1322
firebird	2.0	sugar	1.0	viewing	3.0
khomeini	2.0	sect	1.0	theft	2.2541
everest	2.0	chitin	0.3928	hardship	2.0
spitfire	2.0	jew	0.3043	amplitude	2.0
pennant	2.0	feature	0.2369	ozone	2.0
hurricanes	1.7811	tail-stump	0.231	cost	1.5149
pheasant	1.6885	they	0.1994	councillor	1.459
hawk	1.6533	spine	0.1772	space	1.4516
fighter	1.1218	voice	0.1179	workload	1.4174
tal	1.0682	glue	0.105	turnover	1.3806
plane	1.0616	dispute	0.0955	pain	1.1136
ideas	1.0	plaster	0.0675	revenue	1.0533
mighty-vanned	1.0	glass	0.0635	consumption	1.0455
cannonball	1.0	parent	0.0591	import	1.0096
break		*melt*		*slow*	
war	11.1582	ice	1.022	it	2.2252
dawn	8.9538	ambitions	1.0	rate	1.0063
it	5.9595	snow-wreaths	1.0	follow-up	1.0
scuffle	4.0	solder	1.0	depreciation	1.0
riot	3.7098	pundit	1.0	regeneration	1.0
gathering	2.6699	soldiery	1.0	devine	1.0
they	2.5522	gelatine	1.0	bloodstream	1.0
thief	2.3577	cool	1.0	punk-neutrons	1.0
wave	2.2259	rainbows	1.0	emigrant	1.0
we	2.0005	pudding	1.0	consolidation	1.0
cholera	2.0	1oz	1.0	goalscoring	1.0
dishwasher	2.0	2oz	1.0	frisbee	1.0
rolls	2.0	englishwomen	0.9258	sickle	1.0
talks	2.0	2oz	0.8144	bidding	1.0
hostility	1.9717	snowball	0.732	pace	0.9549

Table 5.2: Re-estimations for six intransitive verbs from the LEV data set model, in descending order of $f(n)p_\theta(c|n)$

Finally, Table 5.3 and Table 5.4 report the induced lexicons for the transitive verbs *increase* and *break*, respectively, that were produced both by the ALL-based and by LEV-based model. Although a certain extent of overlap can be observed between the lists of subject nouns of different data sets, and also between the lists of object nouns, the subjects and objects of the LEV-lexicons appear more *natural* (e.g. less highly infrequent lexical items).

Transitive Verb Subject-Object Re-estimations (ALL-LEV)							
increase (ALL)				*increase (LEV)*			
subject		object		subject		object	
this	57.075	rate	34.9309	this	162.4827	number	26.1274
population	20.4428	number	31.7857	it	133.0675	proportion	17.0542
rate	20.4211	amount	28.055	wage	20.8862	likelihood	13.3363
policy	17.173	risk	17.6933	smoking	20.7101	dislike	12.0
vastly	16.0	pressure	17.3212	carboniferous	18.0	subvention	12.0
maori	14.7666	chance	17.2954	opec	18.0	salinity	11.7615
opec	13.6178	use	14.6785	maori	16.0	probability	10.8327
synthesise	13.0	proportion	14.3757	worldwide	15.7927	1973/4	10.0
authority	12.579	share	14.368	vastly	15.0305	ease	9.6539
exercise	12.0844	case	13.8819	mobility	15.0	disagreement	9.4875
mip-	12.0	system	12.5412	pregnancy	14.0	consequence	9.339
clo	10.5321	those	11.6817	civilian	13.3005	uk	8.6724
turn	10.039	period	11.0501	synthesise	13.0	thromboxane	8.352
court	9.8003	lead	10.8344	charcoal	13.0	use	8.1022
law	8.7012	life	10.3383	unemployment	12.9909	token	8.0

Table 5.3: Re-estimations for the intransitive verb *increase* from both the ALL and the LEV data set model, in descending order of $f(n)p_\theta(c|n)$

Transitive Verb Subject-Object Re-estimations (ALL-LEV)							
break (ALL)				*break (LEV)*			
subject		object		subject		object	
he	19.4197	wilmhurst	10.0	he	99.1905	silence	45.9248
unigram.x	19.391	megastores	8.0	they	45.7079	news	31.6273
maltman	10.0	purdah	5.0	i	29.3255	record	28.6126
dyble	8.1989	pre-tv	3.7132	musgrove	23.0	deadlock	26.4498
fuzzy-wuzzy	8.0	terrortories	3.6432	unigram.x	20.0	stillness	18.0
loughlin	7.7504	seascape	3.2823	signification	19.0	spell	17.2512
they	7.0722	condemned	3.1155	skater	15.1295	promise	16.8541
wiccans	7.0	kulti	3.0602	karen	14.3154	supplicant	16.0
orris	6.0	skitter	2.7396	tiananmen	12.5119	law	14.6091
ambuscade	6.0	ise	2.7374	babe	11.0	commandment	14.0
offspinner	6.0	mezzanine	2.6821	loughlin	11.0	monotony	14.0
trustbuster	5.9596	wilsonville	2.2949	blake	10.4683	fall	11.3201
it	5.8679	gymnast	2.2865	ashton	10.4083	wilmhurst	10.0
nofomela	5.3351	back-row	2.1112	rovers	10.4026	sharon	9.1068
plow	5.2172	clearances	2.0	bolton	10.378	continuity	9.0

Table 5.4: Re-estimations for the intransitive verb *break* from both the ALL and the LEV data set model, in descending order of $f(n)p_\theta(c|n)$

CHAPTER 6
DISCUSSION AND CONCLUSION

In view of the experiment that was set up and carried out up to this point of the study, I share some relevant reflections in the present section, pointing out the achievements and the limits of this work. Observations that will be proposed concern the methodological choices that were adopted and how they contribute to a justified model.

6.1 Observations and Limitations of the Methodology

The present research aimed to set up an application related to the linguistic concept of selectional preference. The methodology that has been applied is directly inspired by the work of Rooth et al. (1999), but has been improved substantially, especially in terms of evaluation. To the best of my knowledge, the combined evaluation system that was built had not been used before in a computational linguistics task related to selectional preference annotation or verb sense induction, and specifically tasks where clustering methods are involved. The main parameter that had to be decided a value for — and with a certain precision, unlike the window of possibilities proposed in Rooth et al. (1999) — was the number of components of the model. A certain robustness of this parameter's estimation has been achieved by weighting out the series of metrics involved. Variational Bayes (VB) has showed being an unsupervised data-driven indicator of the number of active components in a Gaussian Mixture Model (GMM), weighting substantially on the reduced window of candidate values for the parameter in question, given by AIC, BIC, SC, and LLV. At the same time, the estimation of the lower cut-off value for the part of weight distribution to be discarded, remains a point that is worth paying attention to. Visual inspection has revealed being an acceptable method to detect redundant clusters (i.e. with a posterior weight ≈ 0), but a more consistent method may be needed. The very contribution of VB is that even in the case in which the number of classes is known, the number of active components may slightly differ from it. This is illustrated in Figure 6.1, where a VB GMM is fitted on the IRIS data set[1], based on different upper-bound number of components (i.e. 5, 10, 15, 20, 50, 100). Here, the number of classes *should be* 3 (i.e. three types of flowers), but the algorithm does not know the true class label and, independently from that, it identifies all major patterns in the data. Again, the challenge of deciding on a lower-bound threshold remains, which is directly necessary to prune the inactive components in the model.

[1] https://archive.ics.uci.edu/ml/datasets/iris, accessed on July 2nd, 2019

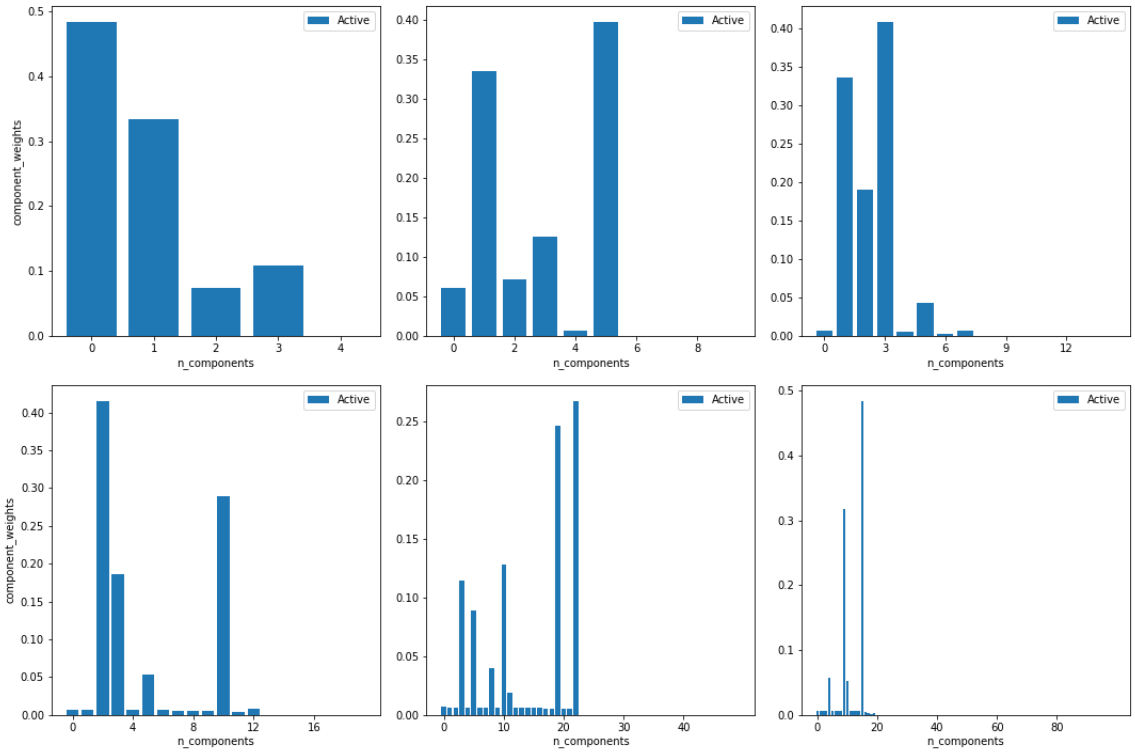

Figure 6.1: Active components for different thresholds and upper-bounds in the IRIS clustering model.

Furthermore, the Levin (1993)-based quantitative analysis of the verb classes in §4.7 are a good adaptation of the method in Sun and Korhonen (2009), especially because of the difficulty of an *a posteriori* labeling. The known limit of this modified purity and weighted accuracy method, is that it is exclusively based on the intransitive verbs and its outcome would potentially be higher for a task that is similar to Sun and Korhonen (2009).

With respect to the pre-processing and feature extraction steps, the tools described in §3.2 have been valid choices, but with some limitations. The lemmatizers used (Smedt & Daelemans, 2012; Loper & Bird, 2002) did not cover highly infrequent lexical elements or noisy data of the BNC. An example is the word *punk-neutrons* in Table 5.2. As mentioned before, this shortage does not influence the clustering process if these unlemmatized elements are the only form of their respective lemma.

An issue related to the parsing is the choice to not consider functional particles, such as prepositions, as strict part of a verb. The result of this can be observed in Table 5.2, where the verb *break* encounters the noun *war* as first element of the induced lexicon for its subject slot. Intuitively:

14. (a) The war breaks *out*.
    (b) *The war breaks.

According to the same intuitions, *break* differs from *break up* and *slow* differs from *slow down*. The question is whether a composed verb should be treated as a whole or not. Although a verb may also have different verb senses without morphologically displaying it, the explicitation of a particle seems to directly convey a different meaning from its individual counterpart, as is visible in Example (13), and should therefore be taken into account.

6.2 Future Research

Given the usefulness of Variational Bayes for the estimation of the number of components, and its limits reported in §6.1, a valid improvement is the use of repulsive processes in the fitting of the model. This technique is explored by Xie and Xu (2019) and Petralia, Rao, and Dunson (2012), among others. The problem that it aims to solve is the redundancy of clusters in a mixture, caused by excessive similarity between the clusters and a lack of a penalty measure that would be able to remove such components. Repulsive processes are a penalizing method for the priors in a Dirichlet process-based technique (such as Variational Bayes) that give rise to the redundant components. Such a technique could contribute to a further step in complete automatic, unsupervised verb classification, by estimating with an even higher accuracy the number of classes to be initialized.

With respect to the clustering algorithms adopted, we can agree on the fact that GMMs work best on the small data set for this task, supported by the property of being semantically more compact than the ALL and F500 data sets. Although K-means started as a back-up being a simpler model than (and part of) Spectral Clustering, it outperformed GMM on the large ALL data set. Therefore, the use of it in future research on a similar task is recommended.

Finally, it would be interesting to apply the same methodology to a different set of verbs. For example ditransitives, although probably low in frequency, or verbs that require a prepositional phrase in one of their argument slots. Also, it would be worth testing whether the distribution of adjuncts over a set of verbs affects the performance of a verb classification task. Such an experiment would show the presence or absence of an ideal, optional *adjunct slot*, at least based on distributional properties. The rationale behind this suggestion is the fact that syntactic properties of adjuncts have already been proven successful for automatic verb classification tasks (Sun et al., 2008).

The set of languages on which a verb classification task has been applied, is still restricted: apart from English, to the best of my knowledge, only Italian (Lenci, 2014), German (Schulte im Walde, 2006) and Chinese (Jinying Chen & Palmer, 2004) have been involved, and all in slightly different tasks. Besides *recruiting* more languages, it could be relevant to apply the same classification and evaluation procedure to multiple languages, ≈*equivalent* corpora and parsing tools permitting. Of course, an interlinguistic analysis is beyond the scope of this thesis, but I hope that my considerations will serve a more elaborated idea.

6.3 Conclusion

The first point of this conclusion is and invocation to a pillar of academic research, namely reproducibility. In the progress of science, previous research becomes fundamental as it forms the building blocks to build further on, improving them step by step or including them in current research. Not only because this thesis builds on previous studies that were at times hardly reproducible, but also since it concludes a research master, the utmost importance of replication is worth a remark. The second consideration is that the trajectory of the present study has finally brought us back to the research question that was introduced in §1:

– "To what extent can the statistical indicator of selectional preference alone, based on a simple clustering algorithm and without the support of external linguistic resources

(taxonomies or dictionaries), lead to the formation of semantically coherent classes that can be used for automatic slot labeling?"

In light of the two experiments have been carried out, we can claim that the clustering has performed well given that it was fully unsupervised: no annotation or labeling, no external resources (evaluation excluded), and no number of clusters defined *a priori*. The LEV data set outperformed the other two data sets, likely because of the verbs in it belong to the Levin (1993) alternation where the object of the transitive is the subject of the intransitive. Although performances were lower for the ALL and F500 data sets, they are in line with the EM-based clustering results reported by Erk (2007) on the same corpus.

The verb classes produced by the models were acceptable, but did not excel when quantitatively compared to Levin (1993), even though that was not required. The same observation is true for the induced lexicons for individual verbs, although no quantitative analysis was performed.

To the research community, this master thesis has contributed in several ways. Precisely, this study offers a view over a specific topic in syntax and semantics, namely the rich field of argument structure and selection, brought to accomplishment through the use of a broad range of data mining tools and machine learning algorithms, the creation of data sets and of an almost full-covering evaluation architecture. Moreover, I moved beyond the founding task of automatic verb classification, by exploiting it and inducing lexicons for individual verbs, which works even for low-frequency verbal entries due to the model's generalization power. The lack of annotated data and sparsity remains a problem unless high amounts of data can be accessed. In any case, annotated external resources have been proven helpful, especially for evaluation.

If selectional preference alone has been shown effective for semantic verb classification, through distributional modeling, there appears to be space for improvement. The goal seems still to be an automatized imitation of the subtle judgments that a human speaker can make about the restrictions and preferences of a verb with respect to its argument configuration. Realistically, the question remains whether all that a speaker knows about a verb is in fact intrinsic of a lexical entry, or not, or even achievable through distributional representations.

REFERENCES

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 17th international conference on computational linguistics-volume 1* (pp. 86–90). Association for Computational Linguistics.

Bergsma, S., Lin, D., & Goebel, R. (2008). Discriminative learning of selectional preference from unlabeled text. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 59–68). Association for Computational Linguistics.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Brew, C. & Schulte im Walde, S. (2002). Spectral clustering for german verbs. In *Proceedings of the acl-02 conference on empirical methods in natural language processing-volume 10* (pp. 117–124). Association for Computational Linguistics.

Brockmann, C. & Lapata, M. (2003). Evaluating and combining approaches to selectional preference acquisition. In *Proceedings of the tenth conference on european chapter of the association for computational linguistics-volume 1* (pp. 27–34). Association for Computational Linguistics.

Carroll, G. & Rooth, M. (1998). Valence induction with a head-lexicalized pcfg. *arXiv preprint cmp-lg/9805001.*

Chen, D. & Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 740–750).

Chen, J. [Jinxiu], Ji, D., Tan, C. L., & Niu, Z. (2006). Unsupervised relation disambiguation using spectral clustering. In *Proceedings of the coling/acl on main conference poster sessions* (pp. 89–96). Association for Computational Linguistics.

Chen, J. [Jinying] & Palmer, M. (2004). Chinese verb sense discrimination using an em clustering model with rich linguistic features. In *Proceedings of the 42nd annual meeting on association for computational linguistics* (p. 295). Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

Dowty, D. R. (1989). On the semantic content of the notion of "thematic role". In *Properties, types and meaning* (pp. 69–129). Springer.

Dowty, D. R. (1991). Thematic proto-roles and argument selection. *language*, *67*(3), 547–619.

Erk, K. (2007). A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 216–223).

Fillmore, C. J. (1968). Lexical entries for verbs. *Foundations of language*, 373–393.

Garvin, P. L. (1962). Computer participation in linguistic research. *Language*, *38*(4), 385–389.

Gildea, D. & Jurafsky, D. [Daniel]. (2002). Automatic labeling of semantic roles. *Computational linguistics*, *28*(3), 245–288.

Gruber, J. S. (1965). *Studies in lexical relations.* (Doctoral dissertation, Massachusetts Institute of Technology).

Hackl, M. (2013). The syntax–semantics interface. *Lingua*, *130*, 66–87.

Hale, K. & Keyser, S. J. (1998). The basic elements of argument structure. *MIT Working papers in linguistics*, *32*, 73–118.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Hindle, D. & Rooth, M. (1993). Structural ambiguity and lexical relations. *Computational linguistics*, *19*(1), 103–120.

Hunter, J. D. (2007). Matplotlib: a 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95.

Jackendoff, R. S. (1972). Semantic interpretation in generative grammar.

Joanis, E., Stevenson, S., & James, D. (2008). A general feature space for automatic verb classification. *Natural Language Engineering*, *14*(3), 337–367.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: open source scientific tools for Python. [Online; accessed ¡today¿].

Jurafsky, D. [Dan] & Martin, J. H. (2014). *Speech and language processing*. Pearson London.

Katz, J. J. & Fodor, J. A. (1963). The structure of a semantic theory. *language*, *39*(2), 170–210.

Kingsbury, P. & Palmer, M. (2002). From treebank to propbank. In *Lrec* (pp. 1989–1993). Citeseer.

Kipper-Schuler, K. (2005). Verbnet: a broad-coverage, comprehensive verb lexicon.

Kipper-Schuler, K., Korhonen, A., Ryant, N., & Palmer, M. (2006). Extending verbnet with novel verb classes. In *Lrec* (pp. 1027–1032). Citeseer.

Korhonen, A., Krymolowski, Y., & Collier, N. (2008). The choice of features for classification of verbs in biomedical texts. In *Proceedings of the 22nd international conference on computational linguistics-volume 1* (pp. 449–456). Association for Computational Linguistics.

Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79–86.

Lapata, M. & Brew, C. (2004). Verb class disambiguation using informative priors. *Computational Linguistics*, *30*(1), 45–73.

Leech, G. N. (1992). 100 million words of english: the british national corpus (bnc).

Lenci, A. (2014). Carving verb classes from corpora. *Word Classes: Nature, typology and representations*, *332*, 17.

Lenci, A. (2018). Distributional models of word meaning. *Annual review of Linguistics*, *4*, 151–171.

Levin, B. (1993). *English verb classes and alternations: a preliminary investigation*. University of Chicago press.

Lewis, M. & Steedman, M. (2014). A* ccg parsing with a supertag-factored model. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 990–1000).

Li, J. & Brew, C. (2008). Which are the best features for automatic verb classification. In *Proceedings of acl-08: hlt* (pp. 434–442).

Lin, D. et al. (1998). An information-theoretic definition of similarity. In *Icml* (Vol. 98, *1998*, pp. 296–304). Citeseer.

Loper, E. & Bird, S. (2002). Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

McCarthy, D. & Carroll, J. (2003). Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, *29*(4), 639–654.

Meila, M. & Shi, J. (2001). A random walks view of spectral segmentation.

Merlo, P. & Stevenson, S. (2001). Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, *27*(3), 373–408.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Mucha, H.-J. & Haimerl, E. (2005). Automatic validation of hierarchical cluster analysis with application in dialectometry. In *Classification - the ubiquitous challenge* (pp. 513–520). Springer.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. In *Advances in neural information processing systems* (pp. 849–856).

Ó Séaghdha, D. (2010). Latent variable models of selectional preference. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 435–444).

Ó Séaghdha, D. & Copestake, A. (2008). Semantic classification with distributional kernels. In *Proceedings of the 22nd international conference on computational linguistics-volume 1* (pp. 649–656). Association for Computational Linguistics.

Osgood, C. E. (1952). The nature and measurement of meaning. *Psychological bulletin*, *49*(3), 197.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pereira, F., Tishby, N., & Lee, L. (1993). Distributional clustering of english words. In *Proceedings of the 31st annual meeting on association for computational linguistics* (pp. 183–190). Association for Computational Linguistics.

Petralia, F., Rao, V., & Dunson, D. B. (2012). Repulsive mixtures. In *Advances in neural information processing systems* (pp. 1889–1897).

Pustejovsky, J. (2016). Lexical semanics. In M. Aloni & P. Dekker (Eds.), *The cambridge handbook of formal semantics* (33fffdfffdfffd64). Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Pustejovsky, J. & Boguraev, B. (1997). Lexical semantics: the problem of polysemy.

Ravin, Y. (1990). Lexical semantics without thematic roles.

Reinhart, T. (2000). The theta system: syntactic realization of verbal concepts. *OTS working papers in linguistics*.

Reinhart, T. (2003). The theta system–an overview. *Theoretical linguistics*, *28*(3), 229–290.

Resnik, P. S. (1993). Selection and information: a class-based approach to lexical relationships. *IRCS Technical Reports Series*, 200.

Resnik, P. S. (1997). Selectional preference and sense disambiguation. *Tagging Text with Lexical Semantics: Why, What, and How?*

Rooth, M., Riezler, S., Prescher, D., Carroll, G., & Beil, F. (1999). Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th annual meeting of the association for computational linguistics on computational linguistics* (pp. 104–111). Association for Computational Linguistics.

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Schulte im Walde, S. (2006). Experiments on the automatic induction of german semantic verb classes. *Computational Linguistics*, *32*(2), 159–194.

Smedt, T. D. & Daelemans, W. (2012). Pattern for python. *Journal of Machine Learning Research*, *13*(Jun), 2063–2067.

Steedman, M. (1987). Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory*, *5*(3), 403–439.

Steedman, M. (2000). *The syntactic process*. MIT press Cambridge, MA.

Sun, L. & Korhonen, A. (2009). Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of the 2009 conference on empirical methods in natural language processing: volume 2-volume 2* (pp. 638–647). Association for Computational Linguistics.

Sun, L., Korhonen, A., & Krymolowski, Y. (2008). Verb class discovery from rich syntactic data. In *International conference on intelligent text processing and computational linguistics* (pp. 16–27). Springer.

Szabolcsi, A. (1992). Combinatory grammar and projection from the lexicon. *Lexical matters*, *1192*.

Xie, F. & Xu, Y. (2019). Bayesian repulsive gaussian mixture model. *Journal of the American Statistical Association*, 1–29.

Yan, D., Huang, L., & Jordan, M. I. (2009). Fast approximate spectral clustering. In *Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining* (pp. 907–916). ACM.

Zapirain, B., Agirre, E., Màrquez, L., & Surdeanu, M. (2010). Improving semantic role classification with selectional preferences. In *Human language technologies: the 2010 annual conference of the north american chapter of the association for computational linguistics* (pp. 373–376). Association for Computational Linguistics.

Appendices

APPENDIX A
DATA

F500 data set ['be', 'have', 'take', 'make', 'include', 'see', 'do', 'become', 'get', 'find', 'show', 'provide', 'know', 'need', 'give', 'contain', 'offer', 'go', 'want', 'hold', 'produce', 'say', 'follow', 'come', 'play', 'bring', 'left', 'set', 'mean', 'require', 'reach', 'involve', 'represent', 'receive', 'used', 'win', 'carry', 'run', 'meet', 'lose', 'form', 'feel', 'write', 'put', 'reflect', 'like', 'open', 'use', 'cover', 'begin', 'face', 'create', 'remain', 'hear', 'describe', 'raise', 'read', 'enjoy', 'buy', 'keep', 'cause', 'turn', 'draw', 'join', 'remember', 'support', 'love', 'spend', 'lead', 'enter', 'develop', 'start', 'wear', 'pass', 'ask', 'pick', 'express', 'send', 'suggest', 'live', 'accept', 'work', 'reveal', 'hit', 'visit', 'add', 'stand', 'seem', 'move', 'leave', 'share', 'affect', 'mark', 'own', 'tell', 'present', 'look', 'announce', 'break', 'suffer', 'build', 'lack', 'feature', 'can', 'seek', 'throw', 'choose', 'increase', 'report', 'fall', 'sell', 'constitute', 'introduce', 'change', 'continue', 'die', 'pull', 'publish', 'walk', 'lay', 'drive', 'reject', 'sit', 'indicate', 'retain', 'comprise', 'understand', 'reduce', 'gain', 'discuss', 'cost', 'miss', 'demand', 'attend', 'attract', 'cut', 'beat', 'allow', 'shake', 'appear', 'examine', 'achieve', 'discover', 'explain', 'establish', 'catch', 'cross', 'return', 'record', 'layer', 'approach', 'serve', 'possess', 'rise', 'maintain', 'launch', 'happen', 'eat', 'stop', 'illustrate', 'watch', 'close', 'control', 'occur', 'strike', 'call', 'place', 'grow', 'ignore', 'end', 'push', 'demonstrate', 'adopt', 'try', 'drop', 'lie', 'acquire', 'issue', 'arrive', 'score', 'pay', 'prefer', 'experience', 'employ', 'consider', 'speak', 'kill', 'encourage', 'learn', 'confirm', 'think', 'identify', 'display', 'lift', 'claim', 'imply', 'remove', 'occupy', 'concern', 'dominate', 'complete', 'stress', 'recall', 'operate', 'exist', 'notice', 'study', 'bear', 'mention', 'round', 'recognise', 'propose', 'deny', 'resemble', 'favour', 'paid', 'combine', 'earn', 'highlight', 'generate', 'extend', 'replace', 'marry', 'will', 'collect', 'incorporate', 'perform', 'hate', 'survive', 'help', '8099', 'avoid', 'sign', 'welcome', 'deserve', 'assume', 'emphasise', 'touch', 'fill', 'prove', 'fly', 'expect', 'manage', 'define', 'exceed', 'shoot', 'recommend', 'pose', 'cite', 'save', 'determine', 'house', 'attack', 'enable', 'head', 'travel', 'fight', 'finish', 'celebrate', 'bore', 'repeat', 'grab', 'obtain', 'dismiss', 'smile', 'supply', 'answer', 'address', 'oppose', 'impose', 'surround', 'influence', 'fit', 'gather', 'point', 'roll', 'conduct', 'encounter', 'teach', 'release', 'boast', 'drink', 'organise', 'handle', 'explore', 'list', 'recognize', 'order', 'arrange', 'destroy', 'observe', 'note', 'reply', 'check', 'promote', 'outline', 'force', 'apply', 'accompany', 'promise', 'abandon', 'match', 'deliver', 'prevent', 'blow', 'vary', 'sound', 'admit', 'pour', 'emphasize', 'improve', 'capture', 'would', 'quote', 'acknowledge', 'wonder', 'treat', 'measure', 'challenge', 'paint', 'defend', 'nod', '=', 'admire', 'permit', 'fear', 'threaten', 'shout', 'thank', 'remind', 'climb', 'last', 'land', 'spot', 'undertake', 'fail', 'embrace', 'stretch', 'reinforce', 'hand', 'provoke', 'pursue', 'the', 'select', 'spread', 'stay', 'imagine', 'emerge', 'undergo', 'rule', 'let', 'review', 'back', 'question', 'kick', 'witness', 'exercise', 'refuse', 'seize', 'founder', 'arise', 'ride', 'plan', 'yield', 'contribute', 'declare', 'trace', 'realise', 'laugh', 'wait', 'advocate', 'test', 'mount', 'forget', 'echo', 'ensure', 'shut', 'convey', 'direct', 'entail', 'lower', 'owe', 'cast', 'count', 'slip', 'limit', 'press', 'command', 'dislike', 'clear', 'prompt', 'design', 'lit', 'analyse', 'exhibit', 'prepare', 'weigh', 'exclude', 'signal', 'wish', 'sweep', 'invent', 'cry', 'resist', 'secure', 'perceive', 'regard', 'knock', 'alter', 'await', 'protect', 'disappear', 'condemn', 'invite', 'overlook', 'link', 'toward', 'separate', 'please', 'enhance', 'detect', 'jump', 'envisage', 'suit', 'steal', 'depict', 'appreciate', 'underline', 'hang', 'justify', 'agree', 'fire', 'defeat', 'distinguish', 'feed', 'greet', 'compare', 'believe', 'encompass', 'unveil', 'inherit', 'predict', 'inspire', 'investigate', 'divide', 'approve', 'sum', 'commit', 'purchase', 'step', 'decide', 'ai', 'burn', 'wipe', 'restrict', 'sang', 'strengthen', 'endorse', 'absorb', 'withdraw', 'hide', 'rub', 'conclude', 'kiss', 'sleep', 'wave', 'resent', 'tear', 'urge', 'shape', 'stimulate', 'hire', 'contact', 'succeed', 'talk', 'precede', 'block', 'respect', 'organize', 'govern', 'embody', 'construct', 'wash', 'sustain', 'restore', 'rang', 'shift', 'arouse', 'trigger', 'sense', 'shrug', 'evoke', 'overcome', 'stick', 'inform', 'matter', 'exploit', 'initiate', 'pack', 'charge']

LEV data set ['bounce', 'drift', 'drop', 'float', 'glide', 'move', 'roll', 'slide', 'swing', 'coil', 'revolve', 'rotate', 'spin', 'tum', 'twirl', 'twist', 'whirl', 'wind', 'break', 'chip', 'crack', 'crash', 'crush', 'fracture', 'rip', 'shatter', 'smash', 'snap', 'splinter', 'split', 'tear', 'bend', 'crease', 'crinkle', 'crumple', 'fold', 'rumple', 'wrinkle', 'abate', 'advance', 'age', 'air', 'alter', 'atrophy', 'awake', 'balance', 'blast', 'blur', 'bum', 'burst', 'capsize', 'change', 'char', 'chill', 'clog', 'close', 'collapse', 'collect', 'compress', 'condense', 'contract', 'corrode', 'crumble', 'decompose', 'decrease', 'deflate', 'defrost', 'degrade', 'diminish', 'dissolve', 'distend', 'divide', 'double', 'drain', 'ease', 'enlarge', 'expand', 'explode', 'fade', 'fill', 'flood', 'fray', 'freeze', 'frost', 'fuse', 'grow', 'halt', 'heal', 'heat', 'hush', 'ignite', 'improve', 'increase', 'inflate', 'kindle', 'light', 'loop', 'mature', 'melt', 'multiply', 'overturn', 'pop', 'quadruple', 'rekindle', 'reopen', 'reproduce', 'rupture', 'scorch', 'sear', 'short', 'short-circuit', 'shrink', 'shrivel', 'singe', 'sink', 'soak', 'splay', 'sprout', 'steep', 'stretch', 'submerge', 'subside', 'taper', 'thaw', 'tilt', 'tire', 'topple', 'triple', 'unfold', 'vary', 'warp', 'blunt', 'clear', 'clean', 'cool', 'crisp', 'dim', 'dirty', 'double', 'dry', 'dull', 'empty', 'even', 'firm', 'level', 'loose', 'mellow', 'muddy', 'narrow', 'open', 'pale', 'quiet', 'round', 'shut', 'slack', 'slim', 'slow', 'smooth', 'sober', 'sour', 'steady', 'tame', 'tense', 'thin', 'triple', 'warm', 'blacken', 'brown', 'crimson', 'gray', 'green', 'purple', 'redden', 'silver', 'tan', 'whiten', 'yellow', 'awaken', 'brighten', 'broaden', 'cheapen', 'coarsen', 'dampen', 'darken', 'deepen', 'fatten', 'flatten', 'freshen', 'gladden', 'harden', 'hasten', 'heighten', 'lengthen', 'lessen', 'lighten', 'loosen', 'moisten', 'neaten', 'quicken', 'quieten', 'ripen', 'roughen', 'sharpen', 'shorten', 'sicken', 'slacken', 'smarten', 'soften', 'steepen', 'stiffen', 'straighten', 'strengthen', 'sweeten', 'tauten', 'thicken', 'tighten', 'toughen', 'waken', 'weaken', 'widen', 'worsen', 'acetify', 'acidify', 'alkalify', 'calcify', 'carbonify', 'dehumidify', 'emulsify', 'fructify', 'gasify', 'humidify', 'intensify', 'lignify', 'liquefy', 'magnify', 'nitrify', 'ossify', 'petrify', 'purify', 'putrefy', 'silicify', 'solidify', 'stratify', 'vitrify', 'americanize', 'caramelize', 'carbonize', 'crystallize', 'decentralize', 'demagnetize', 'democratize', 'depressurize', 'destabilize', 'energize', 'equalize', 'fossilize', 'gelatinize', 'glutenize', 'harmonize', 'hybridize', 'iodize', 'ionize', 'magnetize', 'neutralize', 'oxidize', 'polarize', 'pulverize', 'regularize', 'stabilize', 'unionize', 'vaporize', 'volatilize', 'westernize', 'accelerate', 'agglomerate', 'ameliorate', 'attenuate', 'coagulate', 'decelerate', 'de-escalate', 'degenerate', 'desiccate', 'deteriorate', 'detonate', 'disintegrate', 'dissipate', 'evaporate', 'federate', 'granulate', 'incubate', 'levitate', 'macerate', 'operate', 'proliferate', 'propagate', 'ulcerate', 'vibrate', 'cheer', 'delight', 'enthuse', 'gladden', 'grieve', 'madden', 'obsess', 'puzzle', 'sadden', 'sicken', 'thrill', 'tire', 'weary', 'worry', 'canter', 'drive', 'fly', 'gallop', 'jump', 'leap', 'march', 'race', 'run', 'swim', 'trot', 'walk', 'bang', 'beep', 'blare', 'buzz', 'clack', 'clang', 'clash', 'clatter', 'click', 'hoot', 'jangle', 'jingle', 'ring', 'rustle', 'squeak', 'squeal', 'tinkle', 'twang', 'beam', 'blink', 'flash', 'shine', 'bleed', 'squirt', 'dangle', 'fly', 'hang', 'lean', 'perch', 'rest', 'sit', 'stand', 'swing', 'bivouac', 'board', 'lodge', 'settle', 'shelter', 'asphyxiate', 'choke', 'drown', 'stifle', 'suffocate', 'bleed', 'burp']