Saarland University

Department of Computational Linguistics

Master's Thesis

Automatic Classification of Lexical Aspectual Class Using Distributional and Rule-based Methods

Author:
Liesa Heuschkel
Matriculation: 2531785

Supervisors:
Prof. Dr. Manfred Pinkal
Annemarie Friedrich

January 26, 2016

Abstract

This Master's Thesis introduces various methods for the automatic classification of the aspectual class of verbs in context. The most fundamental distinction of aspectual class is whether a verb is used in a stative or in a dynamic sense. An automatic classification for a verb's aspectual class is needed to interpret the temporal structure of a text or to classify different types of situations expressed by clauses of a text.

Our model uses instance-based, linguistically motivated and distributional features to classify verb instances. Unlike previous approaches, we use clustering on the distributional context of the instances to individualize the previously generalized linguistic indicators more. Our clustered linguistic indicators do not outperform the classic linguistic indicators from previous approaches.

Further, we experiment with different vector similarities, adding additional context to distributional features and constructing new seed sets. We find that adding the context of subject and/or object to the verb vector leads to improvements for a number of highly ambiguous verbs. Our experiments also show that different verbs benefit from different contexts, some only from adding their subject, some from adding the object and some from both.

Previous approaches report the problem of ambiguous verb types, verbs which can be either dynamic or stative depending on the context. Some of these are light verb constructions (LVCs), verbs that are light in meaning and get most of their meaning from their complements. This dependence on the context leads to a need to include context features in the classification system. This thesis investigates the problem and shows that a rule-based approach works better for aspectual classification of LVCs.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Declaration

I hereby confirm that the thesis presented here is my own work, with all assistance acknowledged.

Saarbrücken, January 26, 2016 Liesa Heuschkel

Contents

1 Introduction

Discourse Processing is a promising field in Natural Language Processing (NLP) which is concerned with the underlying structure of texts. One of its tasks is classifying a text's temporal structure or classifying a text's paragraphs as different modes of discourse. Smith (2005) distinguishes the following five Discourse Modes: Narrative, Description, Argument, Information and Report. When classifying these, one needs to first identify their Situation Entities (SE). SE are types of situations which are expressed by clauses of a text.

For the classification of SE, one needs to find out a basic aspect: the aspectual class of verbs (Smith, 2005). To describe events and states, the English language uses verbs in either a *dynamic* or *stative* sense. Events are happenings that cause changes and advance narrative time in discourse. They are characterized by verbs used in a dynamic sense (see example (1)). States are unchanging and stay the same over time, such as opinions, thoughts or properties. They are expressed by verbs used in a stative sense, such as to describe circumstances or properties (example (2).

(1) I am *walking* to the park. dynamic
(2) The flowers *smell* good. stative

There have been some approaches for automatically classifying the aspectual class of verbs, such as Siegel and McKeown (2000) and Friedrich and Palmer (2014a). Friedrich and Palmer's (2014a) classifier uses a number of linguistically informed features such as general linguistic indicators for each verb of how often their instances occurred with linguistic features (see Table 1). They also make use of instance-based features for each instance. Lastly, their classifier utilizes distributional features which consist of similarity scores between the instance to be classified and seed sets of dynamic and stative verbs.

Verb	noSubj	present	past	negation	...
say	0.453	0.321	0.654	0.101	...
make	0.324	0.222	0.814	0.732	...
take	0.102	0.865	0.452	0.233	...

Table 1. Linguistic indicator examples for *say*, *make* and *take*

Our work now extends upon this approach and tries to improve the automatic classification of aspectual class with a number of experiments:

First, we experiment with the linguistic indicator features inspired by Siegel and McKeown (2000). They gather and evaluate these indicators using different Machine Learning methods for the individual indicator combinations. These features indicate how often a verb occurs with features such as the past or present tense or negation. We do not want to include these features as generalized features for each verb type as we would lose data specific to the verbs' different senses. Therefore, we cluster the linguistic indicator feature data for all instances of a verb into multiple clusters and compute indicator features per cluster. Before automatic classification, we compute the automatically assigned cluster for each instance to be classified and use that cluster's linguistic indicator features for our classifier. We use multiple clusters to capture the different senses of each verb. Unfortunately, our results do not show any improvements over the linguistic indicators used by Friedrich and Palmer (2014a). We believe this may be due to incorrect cluster assignments of our instances.

Our automatic classifier utilizes seed sets for computing the similarity between a verb instance and sets of verbs categorized by their aspectual class. We collect and construct a number of seed sets to compare to each other and evaluate if Friedrich and Palmer's (2014a) seed set is the best choice for classification. We show some improvements for a seemingly random selection of verbs over their seed sets but find none that performs consistently better.

We also experiment with additional distributional features that we can use along the ones of Friedrich and Palmer (2014a). We investigate different ways of computing the similarity between the vector of the verb instance and the seed set vectors, such as using the computed average similarity or constructing a centroid vector beforehand and only calculating one similarity. We show that a system using computed average similarities performs slightly better with an accuracy of 72.8, compared to the accuracy of 72.3 of using a centroid vector.

Additionally, we experiment with contextualizing the verb vectors with the verb's subject and/or object using Thater et al.'s (2011) Vector Space Model. How classification can benefit from the context is highlighted by examples (3) and (4).

(3)	Ew, she's *making soup* again!!	dynamic
(4)	This approach does not *make* any *sense*.	stative

Although both examples use the same verb, the context decides whether it is used in a stative or dynamic sense. Incorporating for example the direct object should help with resolving ambiguity and correctly classifying these instances. Our experiments prove that a number of verbs benefit from additional context. Some verbs, like *consider*, benefit from the context of their subject and object. Others, like *accept* and *show*, only benefit from adding their subject to the vector. Another group of verbs, like *fill* and *allow*, benefit from their object. We show that for classifying the aspectual class, the vectors of the verbs in question should be created with the amount of context that is beneficial to the verbs' classification.

The verb *make* in examples (3) and (4) is a light verb. Light verbs occur in light verb constructions (LVCs) which consist of a "semantically bleached" verb and its complements. These LVCs differ from normal verbs in that LVCs get their meaning mostly from their complement. Even though example (3) and (4) use the same light verb they have different meanings. Examples (5) and (6) use a normal verb, the meaning stays the same.

(5)	I'm baking a cake!	dynamic
(6)	They bake their flatbread in a tandoori oven.	dynamic

Classifying the aspectual class of an LVC is especially challenging due to this dependence on context to gain full meaning. A light verb changes its aspectual class depending on its arguments. We examine whether we need a different handling for LVCs than the one we use for non-light verbs. We conduct a corpus study on six light verbs, identifying a number of conventionalized uses for each which in turn leads us to a establishing a rule-based classification system for light verbs. We create 12 regular expressions for five of the six verbs and show improvements of up to 5.1% accuracy.

Next, we will provide some background for the theory of Situation Entities (Smith, 2005) as well as discuss the approach by Friedrich and Palmer (2014a) and other related work in section 2. Section 3 will expand upon experiments centred around clustering linguistic indicator features while section 4 will details our experiments using distributional methods. Section 5 will detail our work on LVCs. This thesis will end with a conclusion in section 6.

2 Related Work

This section introduces the related work relevant for this Masters' Thesis. We provide some background on situation entities (Smith, 2005) and discourse modes (Smith, 2003). Then, we introduce an approach for automatic prediction of aspectual class by Friedrich and Palmer (2014a).

2.1 Background on Situation Entities and Discourse Modes

This section provides information on the motivation for our work. One reason for the classification of the aspectual class of verbs is that it aids the distinction of **situation entities** (SE). Situation entities are individuals, times, concepts or situations introduced by sentences in a text. These SE are addressed via various research questions by the Situation Entities Project[1] at Saarland University. One of their research questions addresses automatic classifying systems for SE. For automatically classifying SE, one must be able to identify their aspectual class. Improving the automatic classification of aspectual class is therefore an important step for this project.

Next, we will introduce some background of SE types. We characterize four main kinds of SE: Eventualities, General Statives, and Abstract Entities (Smith, 2005) and an additional category: Speech-act types (Palmer et al., 2007).

a. Eventualities consist of events, particular states or reports.

(1)	I drank a nice cup of tea.	(Event)
	My tree is blooming.	(State)
	said the farmer about his dog.	(Report)

b. General Statives are Generics and generalizing sentences.
Generics are generalizing statements about objects. They make a statement about all the individuals of the group in question.
Generalizing sentences report about regularly occurring situations related to specific individuals. These are not events or states but patterns of situations. Other names include gnomic, dispositional, general or habitual situations. Regularity words like *sometimes, always* or *never* are used frequently which can also be used as a test for identifying Generalizing Sentences. This is done by adding such a word and checking whether the meaning or syntax has changed.

(2)	German men wear trekking sandals.	(Generic)
	The neighbour's dog always barks.	(Generalizing)

[1] http://www.coli.uni-saarland.de/projects/sitent/page.php

c. Abstract Entities can be facts or propositions. The former are known truths whereas the latter are beliefs. These two situations are different from the previous ones in that they are not spatially or temporally located.

(3)	I know that Germans love sparkling water.	(Fact)
	I believe the Asian grocer is raising its prices.	(Proposition)

d. Speech-act Types can be either questions or imperatives.

(4)	Did you wash the dishes?	(Question)
	Run, Forest, run!	(Imperative)

When trying to distinguish Situation Entities one can make use of three features (Friedrich and Palmer, 2014b): the type of the main referent, the fundamental aspectual class, and habituality.

The **main referent**, or central entity, of a clause is usually the subject of the sentence. It occurs as a noun phrase and can be determined by the question *What/Who?*. It can either be specific (5) or generic (6). The main referent can be entities, groups of entities or organizations as well as a situation or instantiation of a concept (7).

(5)	*The doctor* examines a patient.	(specific referent, STATE)
(6)	*Cucumbers* are green.	(generic referent, GENERIC SENTENCE)
(7)	It's odd *that you reacted badly to the food.*	(specific, EVENT)
	Today's weather was cold and snowy.	(specific, STATE)

The **fundamental aspectual class** is an essential feature for the classification of verbs. This is the feature that this thesis wants to classify. Section 2.2 will go into more detail on aspectual classes in general. Verbs can be distinguished by the following aspectual class features:
Dynamic: A verb and its arguments describe that something is happening.
Stative: Nothing is happening, only properties of the main referent are introduced.
Both: Dynamic and stative interpretations are both possible.

(8)	I'm going to take a shower.	(dynamic, STATE)
	I write letters.	(dynamic, GENERALIZING SENTENCE)
	I ate a burger.	(dynamic, EVENT)
	My sister has black hair.	(stative, STATE)

Habituality refers to the frequency of a state or event. Something *episodic* occurs only once. Regularly occurring states or events are called *habitual.* Habitual sentences do not change their meaning if one adds a frequency adverbial.

SE Type	Main Referent	Aspectual Class	Habituality
Event	specific generic	eventive	episodic
State	specific	stative	static
Generic Sentence	generic	eventive stative	habitual stative, habitual
Generalizing Sentence	specific	eventive stative	habitual
General Stative	specific generic	eventive	habitual

Table 2. Distinction between Situation Entities

(9) My dad had a flat tire yesterday. (episodic, EVENT)
I go to work every Wednesday till Friday. (habitual, GENERALIZING SENTENCE)
Goats eat old fruit. (habitual, GENERIC)
Goats usually eat old fruit. (habitual, GENERIC)

With these three features one can distinguish more easily between the different Situation Entities. Some are easier to classify than others: Facts, Propositions, Questions and Imperatives are easy, whereas others are more difficult. Table 1 shows an overview of which SE occur with which features.

Classifying the aspectual class of verbs does not only aid with the identification of SE but also with interpreting temporal information. The latter is incremental for distinguishing **Discourse Modes** (DMs). DMs are different types of text passages. According to Smith (2003), such a passage can be in the form of a Narrative, Description, Argument, Information, or Report. DMs are found in texts of all genres unless they are highly scripted and without variation.

Smith (2003) describes DMs as a radical extension of the temporal aspectual notion. Discourse Modes are characterized by aspectual and spatio-temporal features: a) the type of SE they introduce and b) the kind of text progression. No DM employs only certain Situation Entities, which makes identification difficult, although it seems that each DM prefers certain SE types.

The modes Argument and Information progress in an atemporal way, their SE are mostly non-dynamic. They progress with the locations that are being introduced in the text. The DMs Narrative, Description and Reports are temporal with passages progressing over the span of time and space. Table 3 lists the five Discourse Modes and the SE they mainly employ, as well as their mode of progression (Smith, 2003).

Mode of Progression	Discourse Mode	Primary SE	Other SE
Temporal	Narrative	specific Events, States	-
	Description	Events, States	ongoing Events
	Reports	Events	States, General Statives
Atemporal	Argument	Facts, Propositions	General Statives
	Information	General Statives	-

Table 3. Discourse Modes and Situation Entities

2.2 Aspectual Class of Verbs

This section will further explain the main topic for this thesis. As mentioned before, situation entity types can be classified by the fundamental aspectual class (also: Aktionsart) of the sentence.

This kind of classification dates back to Aristotelian time. Aspectual properties are temporal. Vendler (1967), as well as Dowty (1979), recognize four classes: States, Activities, Achievements, and Accomplishments.

Each state has a combination of the following features: dynamic-static, telic-atelic and durative-instantaneous. Events are **dynamic**, whereas states are **static**. Dynamic events occur in stages over the span of time whereas states are constant in time. **Telic** events have a certain endpoint or a goal. **Atelic** events can end at any time. **Durative** events have a set (longer) duration and **instantaneous** events only consist of one stage at a point in time.

Here we provide some short explanations with examples:

States do not occur in progressive tenses. States describe the state of an individual or object at a particular time.

(11) She knows my address.
I love her.
The laundry is on the drying rack.

Activities occur over time without an end result. They do not allow adverbial prepositional phrases with *in*.

(12) I am writing for days and days.
The bus is on its way to university.
*The dog listens for the burglar in 5 minutes.

Achievements do not occur in progressive tenses but happen at a particular moment. They have some kind of end result or achievement.

Class	Dynamic	Telic	Durative
States	-	-	+
Activities	+	-	+
Achievements	+	+	-
Accomplishments	+	+	+

Table 4. Aspectual Classes and their Features

(13) The thesis was finished.
The squirrel had hid the nut in the perfect spot.
The young woman won the lottery drawing.

Accomplishments are durative with some kind of end result.

(14) We moved flats.
My grandma is baking cookies.
UPS is delivering a package.

A summary of these features can be seen in table 4.

While Vendler (1967) and Dowty (1979) distinguish between four classes, Bach (1986) makes the basic distinction of eventualities into **states** and non-states (see Figure 1). These non-states consist of **processes** and **events**.

States are classified as either static or dynamic. He asserts that only dynamic states can occur in progressive tense (see example (15)). This is in contrast to our definition of states which can only be of the aspectual class stative.

(15)	*She is knowing the riddle.	static
	*I am loving my wife.	static
	They are lying on the ground.	dynamic

Processes consist of verbal predicates such as example (16).

(16)	The geese walk slowly.	process
	The kids push the shopping cart.	process

Events can be protracted (see (17)), meaning they are lasting for a long time, or momentaneous. For the distinction of momentaneous events, Bach (1986) recognizes culminations (example (18)), which are the highest point of some kind of build-up, and happenings (example (19)), which are characterized by a beginning and an end happening more or less at the same moment.

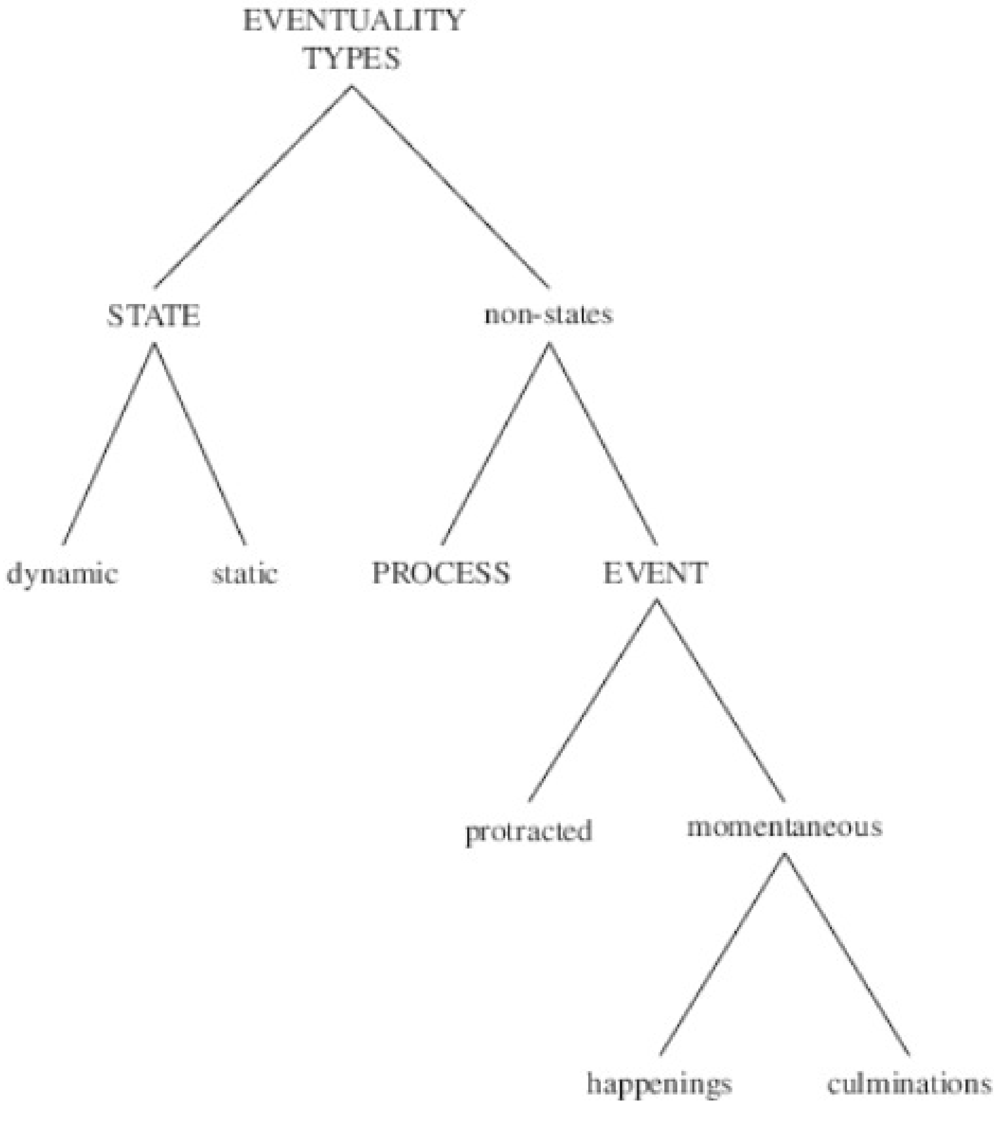

Figure 1. Bach's (1986) Classification of Verbs

(17)	I am building a house.	protracted event
	We walk to the store.	protracted event
(18)	My dog died.	culmination
	The mountaineer reached the top.	culmination
(19)	The store manager noticed spilled milk.	happening
	They recognized each other at one.	happening

As mentioned before, this work aims at classifying verbs by their aspectual class state, using the features dynamic and stative. This concurs with Bach's (1986) distinction of states vs. non-states and Vendler's (1967) class State.

Predicting a verb's aspectual class in context can help with many NLP tasks. The aspectual class is one of the features needed for distinguishing between SE (Smith, 2005). It is also important for interpreting the temporal structure of a text. Moens and Steedman (1988) propose a new formal structuring of linguistic events that does not strictly rely on a temporal order but rather on an order based around a nucleus. This nucleus is a structure consisting of a culmination embedded between a preparatory process and a consequent state. The authors believe that categories such as aspects or adverbials change the temporal or aspectual class of verbs.

The next chapter will introduce a state-of-the-art classification system for aspectual class.

2.3 Automatic Prediction of Aspectual Class of Verbs in Context

Friedrich and Palmer (2014a) describe an approach of automatically predicting the aspectual class of a verb in context. The aspectual class is an attribute concerning the kind of event that is described by the verb and its arguments. An aspectual class can either be a state, a process, or an event (Siegel and McKeown, 2000). The authors investigate whether a verb is being used in a stative or a dynamic way.

Most verbs are usually either predominantly dynamic or stative (see (20)). Some verbs are more flexible and can be read as both ways depending on the context (21).

(20)	I like food.	stative
	The flower grows fast.	dynamic
(21)	I have a bike and a car.	stative
	I'm having a cup of tea.	dynamic

The authors use several data sets:

- Verb type seed sets constructed from the LCS (Lexical conceptual Structures) database (Dorr et al., 2001). There are three seed sets: *dynamic*, *stative* and *both*.

They contain words with entries in the LCS database as either only *dynamic* or only *stative*. If a word has entries for both classes it is used for the *both* seed set.

- Asp-MASC which consists of full texts from MASC (Ide et al., 2010), annotated with the aspectual class of the main verb.

- Asp-Ambig which consists of sentences for 20 frequent verbs. The sentences are annotated with their aspectual class.

Unlike previous approaches (see Siegel and McKeown (2000)) the authors use a combination of linguistic indicator, distributional and instance-based features.
Linguistic indicator features are a attribute properties that can be gathered from a large corpus (Siegel and McKeown, 2000). Friedrich and Palmer (2014a) count how many times a verb type occurs with certain indicators. These indicators include tense, negation, particle, adverb type, PP type and more. For the **distributional features** they obtain context vectors for each verb type using Thater et al.'s (2011) Vector Space Model and compute the average of the cosine similarities between all instances of a verb and each of the verb types from the three verb seed sets. These aforementioned seed sets consist of verbs that have entries in the LCS database in the dynamic or stative aspectual class category or in both.
The **instance-based features** are extracted from the clause the authors want to classify. These features include tense, progressive, perfect, voice and information about the dependent of the verb.

The authors evaluate their feature sets in four experiments. At first, they investigate how well their feature sets perform for seen verbs. No feature set is better than simply memorizing the class the verb type was observed as most. Secondly, the performance on unseen verbs is evaluated. A combination of Linguistic indicator and Distributional features is found to work best. Thirdly, one- and multi-label verbs are investigated. The baseline performs best for one-label verbs whereas Instance-based features turn out to be essential for classifying multi-label verbs. Finally, the classification for verbs with an ambiguous aspectual class is evaluated. These verbs have to be dealt with separately as a type-based classification would just assign the dominant sense to them. Using the verb lemma and Instance-based features turns out to work best. Depending on the verb type, adding Linguistic indicator and Distributional features improves the accuracy.

Friedrich and Palmer (2014a) show that it is possible to predict the aspectual class of verbs by using labelled training data and instance-based features. Type-based features are essential for a successful classification of unseen verbs and ambiguous verb types.

3 Improving Classification of Aspectual Class Using Clustering of Linguistic Features

In this chapter we try to improve the linguistic indicator features which we use for classification. These features are generalized per verb type, which makes it impossible to reach an accuracy better than the majority class. We test the hypothesis that computing these features based on clustered distributional data of the instances will lead to more individual feature groups and therefore to an improved system.

Next, this chapter will expand on the problem at hand, used methods, corpus data and statistics, our experiments and then finally a summary.

3.1 Problem

The linguistic indicators by Siegel and McKeown (2000) which we use for aspectual class classification are averaged over different usages of the same verb. Table 5 shows some example data for the verb *make*. Here, four instances of *make* are shown with the extracted occurrences for four linguistic indicators (for-prepositional phrase, future, present and negation). In the bottom row are the final linguistic indicators for this verb, a computed average of all instances. These features are collected by extracting them from Gigaword sentences for every verb's occurrences.

verb	for_pp	future	present	negation	...
$make_1$	1	0	1	0	...
$make_2$	0	0	1	0	...
$make_3$	0	1	0	1	...
$make_4$	0	0	1	0	...
make	0.25	0.25	0.75	0.25	...

Table 5. Linguistic Indicators for *make*

Since all usages are grouped together, information related to word senses is lost. This makes building a classifier with a better accuracy than just using the majority class difficult. With our last experiments we try to improve these linguistic indicators by first clustering the verb instances and then using the linguistic indicators per cluster for classification.

3.2 Corpus Data and Statistics

In this subsection we detail the corpora used for our clustering experiments.

3.2.1 Sentences from Gigaword AFE/XIE

We extract all sentences with occurrences for 19 verbs from Gigaword AFE/XIE (Graff and Cieri, 2003). Gigaword is a large text corpus consisting of English newspaper articles. Here, we use the AFE (Agence France Press English Service) and XIE (The Xinhua News Agency English Service) parts. The number of all these instances can be seen in table 6.

3.2.2 Asp-Ambig

This corpus consists of 138 sentences for 20 frequent verbs (Friedrich and Palmer, 2014a). The sentences were randomly extracted from the Brown corpus and annotated. The two annotators had to choose one of three labels (*stative/dynamic/both*) after being shown the whole sentence. The final corpus consists of 2760 instances. κ is 0.6.

3.2.3 Asp-Ambig_MASC_Wiki

We construct this corpus by combining the following corpora:

- **Asp-Ambig**: This corpus is described in 3.2.2.
- **Asp-MASC**: This corpus consists of texts from the Manually Annotated Sub-Corpus (MASC) annotated with various Situation Entity information, such as the aspectual class (Friedrich and Palmer, 2014b).
- **Fulltext_Wiki**: The Fulltext_Wiki corpus consists of 9514 sentences extracted from 78 full text articles from Wikipedia. We present each sentence separately to our 3 annotators and let them choose one of three labels (*stative/dynamic/can't decide*). Cohen's κ is 0.65. The gold standard was chosen by majority vote.

We only choose instances labelled as stative or dynamic. The final corpus consists of 32,657 instances for 2494 verb types.

Verb	Giga AFE/XIE
accept	48212
allow	97691
appear	47331
bear	22336
carry	100874
come	222430
consider	64343
cover	41480
feel	33724
fill	8274
find	125220
follow	167025
hold	273697
look	60040
make	408738
meet	189563
show	123090
stand	52283
take	421771
ALL	2508122

Table 6. Number of Extracted Instances from Giga AFE/XIE

3.3 Method

3.3.1 Collecting Bag-of-Words Data

For this row of experiments, we use instances from Gigaword AFE/XIE. We extract all sentences for our 20 verbs. Since we already receive our sentences lemmatized, we simply compute a bag-of-words by counting the occurrences of the verb with its 10 neighbor words on the left and on the right side. We then construct one BOW per verb by using all the verb's neighboring words as dimensions.

3.3.2 Singular Value Decomposition

This bag-of-words data contains many zeros and is high dimensional. Therefore we use truncated singular value decomposition (a faster version of SVD, from Python Sklearn package (Pedregosa et al., 2011)) before clustering. By using Truncated SVD we perform a dimensionality reduction on our huge data set while keeping the relationships between our instances.

Singular value decomposition (SVD) is the process of decomposing a real or complex matrix A into multiple matrices. It is used as a intermediate step in many algorithms, most often to reduce the number of dimensions in data (Baker, 2005).

SVD factorizes an m x n matrix A into:

$$A = U\Sigma V^T$$

U is, as the name suggests, a unitary matrix, Σ is a matrix of m x n shape with non-negative real numbers on its diagonal axis, and V^T is the transposed version of V, an n x n matrix (see Figure 2).

Also, $U^T U = I$ as well as $V^T V = I$. While eigenvectors of AA^T make up the columns of U, V's columns are eigenvectors of $A^T A$. Σ's values are zero except for its diagonal, which consists of square roots of the eigenvalues from U or V.

By decomposing our original matrix A into these other matrices, we have a way to reduce the dimensions of our original data while finding its best approximation. SVD is also used to discover hidden relationships between the data points and find the dimensions with most variance for the data points.

We then continue working on the dot product between U and the diagonal of Σ. Compared to the original matrix, this new matrix has fewer dimensions but still encodes the

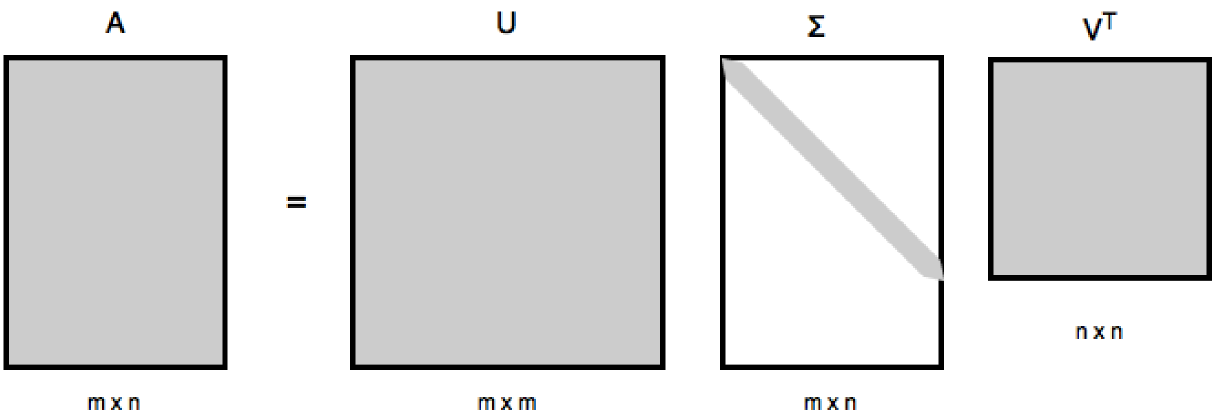

Figure 2. Singular Value Decomposition

relationships between our instances. We experiment with SVD by performing two experiments with different parameters: one with two dimensions for the new matrix and one with 100 dimensions.

3.3.3 Clustering

We then use this new matrix of dimensionally reduced data for clustering. A cluster analysis, or **clustering**, groups similar objects together and assigns them to groups, so called clusters. This task is used in many fields, such as Machine Learning, Data Mining and Information Retrieval.

We use k-means clustering (MacKay, 2003). The algorithm works on data points, also called observations, in a vector space. Our observations are each verb's instances and their distributional data. K-means tries to group similar verb instances together based on their position in the vector space which is determined by this distributional data.

K-means starts by initializing cluster centres. The number of these centres is usually pre-determined, but there are also ways of determining the perfect number of clusters. The initialization of the clusters is an important step. The initial clusters will have a big impact on what the later clusters will look like. There are different methods on how to determine the initial clusters: one can put them in a straight (vertical, horizontal or diagonal) line, evenly spaced from one another, one can pick data points from the data set or points can just be set randomly.

After initialization, the algorithm always does one of two things: it either assigns a data point to a cluster or it moves the cluster centre.

The data points are assigned to a cluster by computing the distance from each point to each cluster centre. The distance measure commonly used is Euclidean distance. The

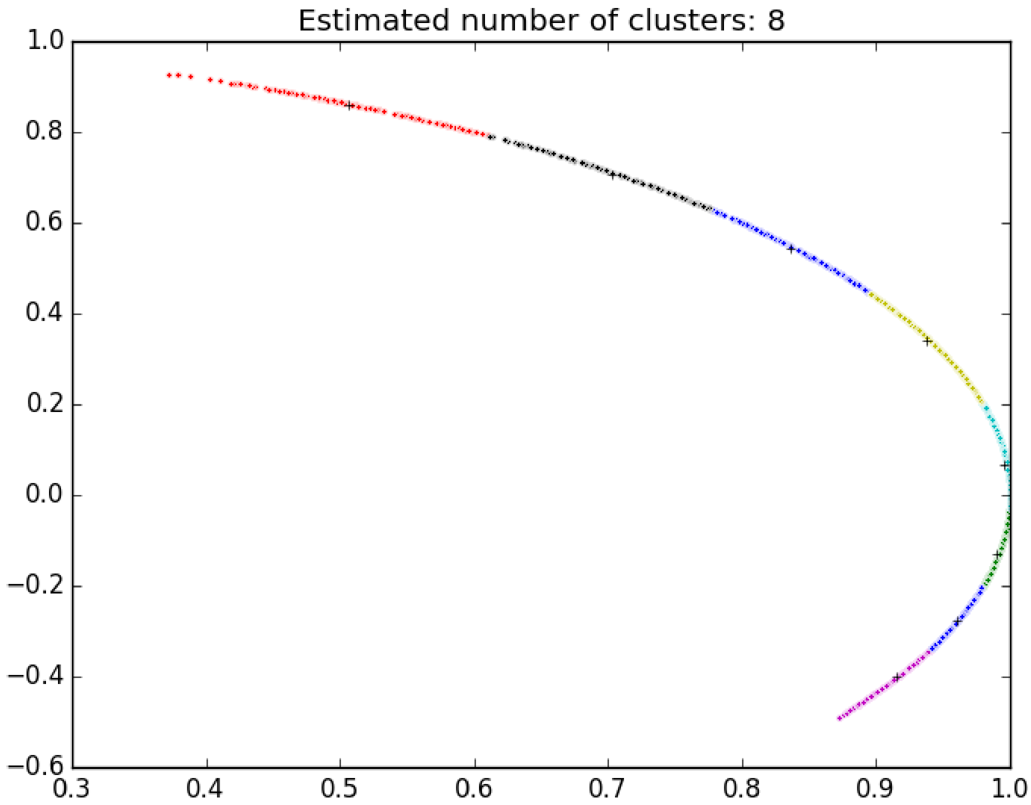

Figure 3. Clusters for Verb *look*

point will then be assigned to the nearest cluster. After each total iteration over all points, the cluster centres will be recalculated. This is achieved by finding the mean of all the instances in each cluster. Then the assignment starts again using the newly computed cluster centres. When no change in assignments occurs, the algorithm converges.

We use the Mini Batch k-Means implementation of the Python Sklearn package (Pedregosa et al., 2011). Mini Batch k-Means is a variant of k-Means with a faster computation. This is achieved by the algorithm working only on a subset of the data, thereby reducing the number of distance computations needed for computing the clusters (Béjar Alonso et al., 2013).

First, we perform clustering with a fixed number of 8 clusters on the first 40,000 instances. For all additional instances, we only predict their cluster. A picture of the cluster assignments for the verb *look* can be seen in Figure 3.

After clustering, for each verb's instance we receive an assignment to one of the 8 clusters for that verb. Exemplary data for the verb *look* can be seen in Table 7.

Cluster	Example Sentences
0	[...] their leader need look no further than his mirror
	[...] begin to look for partner elsewhere [...]
	they look suspiciously at the three western reporter [...]
	[...] we be look at a 15 billion dollar surplus [...]
1	she look poise to dominate the final [...]
	first-quarter growth look even stronger than we have assume [...]
	[...] sift through the waterfront rubble look for more potential victim [...]
	we have to look at ourselves [...]
2	they both look forward to a close relationship burns say
	[...] so we be look at up to 10 year before we get some result
	we be look more at logistic support albright say on cnn television on sunday
	but i define significance by look [...]
3	the agreement of 1993 be not what we be look for anymore [...]
	the poles be look to dudek to keep the home attack at bay
	[...] i 'll look for another way to serve my country
	[...] and be look to make up for the disaster [...]
4	[...] look to the football team as an escape from their daily problem
	[...] ministry need to look at the whole issue of tariff and surtax ratio
	[...] i have to say it look like terrorism plain and simple boucher say
	[...] if you keep perform we will look at you
5	[...] the political future [...] look increasingly uncertain
	india never look like repeat their 1975-76 feat [...]
	[...] indonesia should look at make [...] islands free trade zone as well
	[...] nigeria be look forward to the development of friendly cooperation [...]
6	[...] he would have look at i with those puppy eye and try to keep i go
	the indian skipper look for a big knock [...]
	[...] i would like to look for cooperative relation [...]
	lenin avenue [...] look as if it have be hit by an earthquake
7	[...] yeltsin say look grim-faced and thump the table
	[...] the government be look for they [...]
	and we look forward to a day [...]
	fbi and local investigator be look into the october 9 crash [...]

Table 7. Cluster Assignments for Verb *look*

Linguistic Indicator	Explanation
neg	verb is negated
adv_evaluation	adverb of class evaluation (alright, badly, ...)
noSubj	no subject
perfect	verb tense perfect
past	verb tense past
present	verb tense present
adv_temporal	adverb of class temporal (again, always,)
adv_continuous	adverb of class continuous (constantly, endlessly, ...)
progressive	verb tense progressive
fut	verb tense future
in_pp	verb used in in-prepositional phrase
prt	particle
for_pp	verb used in for-prepositional phrase
freq	frequency
adv_manner	adverb of class evaluation (accurately, curiously, ...)

Table 8. Descriptions of Extracted Linguistic Indicators

3.3.4 Collecting Linguistic Indicators

We then use this cluster assignment to compute the linguistic indicator features. For each instance, we extract linguistic indicators from Giga AFE/XIE according to Friedrich and Palmer (2014a). The descriptions for these linguistic features can be seen in Table 8.

For each of the eight clusters, we add the linguistic indicators of the cluster's assigned instances together. We normalize by dividing each feature value by the number of instances per cluster. Linguistic indicator data per cluster for the verb *look* can be seen in Table 9.

Cluster	fut	neg	past	perfect	present
0	0.000477	0.000477	0.004350	0.000565	0.005286
1	0.000221	0.000783	0.003303	0.000493	0.007991
2	0.000321	0.000429	0.003788	0.000535	0.006452
3	0.001052	0.000707	0.005480	0.000523	0.004310
4	0.000438	0.000479	0.004023	0.000505	0.005944
5	0.000273	0.000461	0.003719	0.000606	0.006735
6	0.000381	0.000512	0.003861	0.000568	0.005952
7	0.000783	0.000626	0.004652	0.000511	0.004731

Table 9. Excerpt of Linguistic Indicator Features per Cluster for Verb *look*

3.4 Experiments

3.4.1 Comparing LingInd with LingIndCluster

In this section, we evaluate our newly computed features. Our experiments compare the performance of a classifier using the linguistic indicators by Friedrich and Palmer (2014a) and the new linguistic indicators we computed via clustering in the last section.

Our training corpus is Asp-Ambig_MASC_Wiki, which consists of 32,657 instances with the linguistic indicators by Friedrich and Palmer (2014a). We use the Asp-Ambig corpus as a test file. We only use a subset of 2760 instances which are annotated as *stative/dynamic*. As a baseline for each verb, we train on our training corpus with the usual linguistic indicators but withhold the instances for that verb. We then classify our test data using the same LingInd with a RandomForest classifier.

As a comparison, we prepare our test file for the second classification: for each verb, we assign its test instances from Asp-Ambig to one of the verb's 8 clusters. We then use the assigned cluster's linguistic features for classification. We train again on the Asp-Ambig_MASC_Wiki with Friedrich and Palmer's (2014a) LingInd. As mentioned before, we use different SVD parameters as well: SVD with 2 dimensions (LingInd-Cluster-2-SVD) and SVD with 100 dimensions (LingInd-Cluster-100-SVD).

Both classification experiments are done for an "unseen verb" case. While training on the training data and classifying the test data we withhold the verb lemma feature. As mentioned before, we also exclude the instances for the verb type that we are classifying from the training data.

Classification results can be seen in terms of accuracy in Table 10. In this evaluation, LingInd-Cluster-2-SVD never once improves over the majority class. The classifiers perform differently depending on the kind of linguistic indicators they employ. Friedrich and Palmer's (2014a) seems to perform best, reaching the same accuracy as the majority class for 13 of the 19 verbs. For the other cases the classifier seems to classify all instances as the minority class.

3.4.2 Cluster Analysis

The evaluation of our classification systems using the clustered features does not show any improvement. Therefore we conduct an analysis of the cluster assignments.

We compute the cluster purity for each verb. The values for LingInd-Cluster-2-SVD and LingInd-Cluster-100-SVD can be seen in Table 11, as well as a random cluster assignment. The calculated purity shows relatively similar results per verb for the different cluster

Verb	DYN	STAT	Maj. Class	LingInd	LingInd-Cluster-2-SVD	LingInd-Cluster-100-SVD
meet	112	0	100.0	**100.0**	100.0	96.4
make	126	1	99.2	**99.2**	59.1	91.3
feel	2	127	98.4	1.6	1.6	**20.2**
come	117	2	98.3	**98.3**	42.9	20.2
take	118	12	90.8	**90.8**	86.2	56.9
accept	95	10	90.5	**90.5**	90.5	90.5
find	102	11	90.3	**90.3**	90.3	90.3
stand	12	111	90.2	9.8	9.8	**19.5**
follow	48	9	84.2	**84.2**	84.2	84.2
carry	76	24	76.0	**76.0**	76.0	76.0
cover	29	66	69.5	**30.5**	30.5	30.5
show	73	33	68.9	**68.9**	68.9	68.9
fill	64	30	68.1	**68.1**	51.1	63.8
allow	56	27	67.5	**67.5**	34.9	32.5
bear	64	31	67.4	**67.4**	48.4	48.4
look	77	44	63.6	36.4	**63.6**	**63.6**
appear	51	70	57.9	**42.1**	42.1	42.1
hold	26	34	56.7	**43.3**	43.3	43.3
consider	70	59	54.3	**54.3**	54.3	54.3

Table 10. Classification Accuracy using Clustered Linguistic Indicator Features

Verb	Maj. Class	Random Cluster	2-SVD-Cluster	100-SVD-Cluster
meet	100.0	0.8116	0.8116	0.8116
make	99.2	0.9130	0.9130	0.9130
feel	98.4	0.9203	0.9203	0.9203
come	98.3	0.8478	0.8478	0.8478
take	90.8	0.8551	0.8551	0.8551
accept	90.5	0.6884	0.6884	0.6884
find	90.3	0.7391	0.7391	0.7391
stand	90.2	0.8043	0.8043	0.8043
follow	84.2	0.6377	0.5870	0.6014
carry	76.0	0.5507	0.5580	0.5507
cover	69.5	0.4783	0.5145	0.4855
show	68.9	0.5362	0.5435	0.5580
fill	68.1	0.4783	0.4783	0.4928
allow	67.5	0.5145	0.4783	0.4710
bear	67.4	0.5217	0.4928	0.5580
look	63.6	0.5960	0.5725	0.6014
appear	57.9	0.5435	0.5362	0.5797
hold	56.7	0.5580	0.5870	0.5652
consider	54.3	0.5435	0.5362	0.5435
AVG	78.5	0.6599	0.6560	0.6625

Table 11. Cluster Purity Values

assignments. The average cluster purity for 100-SVD-Cluster is just slightly higher (0.01) than the random cluster assignment. The cluster purity for the 2-SVD-Cluster is 0.01 lower than the random cluster.

Since our cluster assignment is hardly more precise than a random assignment, we believe this to be the cause for the lack of improvement for our classifier. It seems like either classifying instances based on BOW or the number of dimensions for SVD is insufficient for cluster assignments.

3.5 Summary

In this section, we discuss our approach on improving the features used in classifying the aspectual class of verbs. We use parts of Gigaword to collect sentences for 19 verbs. We compute distributional information for each instance and cluster each verb's instances into one of eight clusters. We extract linguistic indicators according to Friedrich and Palmer (2014a) and compute each cluster's linguistic indicator features via the cluster assignments for each verb's instances.

We classify with Friedrich and Palmer's (2014a) linguistic indicator features and with the newly computed features. The accuracy of the classifier with the old features is not as good as the majority class, but reaches the same accuracy in the majority of cases. The classifiers with clustered linguistic indicators do not perform as well as either of them. We conduct a cluster analysis and show that our computed clusters show the nearly same cluster purity as clusters with randomly assigned instances.

In the future, we would like to improve upon this clustering approach by further examining SVD and the clustering algorithm and by experimenting with different parameters. Clustering not only on distributional information but also on the linguistic indicators for each instance is another idea to explore. This however is not possible in the time frame of this thesis.

4 Improving Classification of Aspectual Class using Distributional Methods

This section details our experiments of using distributional methods to compute further features for the classification of aspectual class. First, we will explain our reasoning for these experiments, some related work, the experiments and finally a summary.

4.1 Problem

Friedrich and Palmer's (2014a) classifier relies mostly on grammatical features for classification such as their linguistic indicators and instance based features. Their instance-based features add a bit of the verb's context but only in the form of linguistic information such as tense information or the verb's POS-tag.

We investigate whether classification can be improved by adding more context in the form of distributional features. Example (1) shows how two instances can have the same grammatical features but are categorized by different aspectual classes. If one would include the context (in this case *bread* and *sense*), one could more easily classify the instances.

(1)	She is making bread.	dynamic
	She is making sense.	stative

4.2 Background

Vector space models (VSMs) represent words as vectors in semantic space. They use a word's context to model its vector. The vector's elements consist of co-occurrence data, meaning how often the target word occurred with other words in a window of size n. A window of size n means n neighbouring words to the left and n neighboring words to the right of our target word.

An example of a co-occurrence matrix would be Table 12. It depicts how often the words *kitchen, dog* and *cat* occur with other words. This co-occurrence data can then be used to construct word vectors such as the vector for *dog*:

$$\vec{v}_{dog} = [0, 6, 7, 10]$$

One can compute the similarity of two words by comparing their vectors using cosine similarity. If two word vectors are pointing away from each other, they are opposite

	kitchen	dog	cat
oven	7	0	0
food	10	6	7
vet	0	7	3
pet	1	10	8

Table 12. Co-occurrence matrix of a fictional corpus

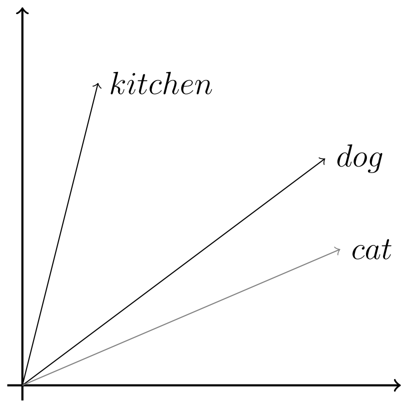

Figure 4. Words being represented as vectors

of meaning. The more similar the direction of the two vectors, the more similar their meaning (see Figure 4). VSMs are easy and fast to implement and work automatically, as well as unsupervised.

Thater et al. (2011) This model is the basis for the distributional experiments used in this thesis. Thater et al.'s (2011) VSM provides methods for representing words as vectors, contextualizing these vectors and calculating their similarity using a range of measurements.
We use this model specifically for its capability of contextualizing our verb vectors using syntactic relations since we believe that the classification would benefit from adding additional information such as the verb's object or subject. Thater et al.'s (2011) VSM takes the semantic similarity information from the verb's local syntactic context to construct co-occurrence data. This data is then used to reweight the vector's components. Figure 5 show how contextualizing a verb vector with its direct object shifts the vector closer to its real meaning.
Thater et al.'s (2011) model outperforms existing systems' precision by 6% for paraphrase ranking and outperforms the word sense disambiguations task's baseline by 3%.

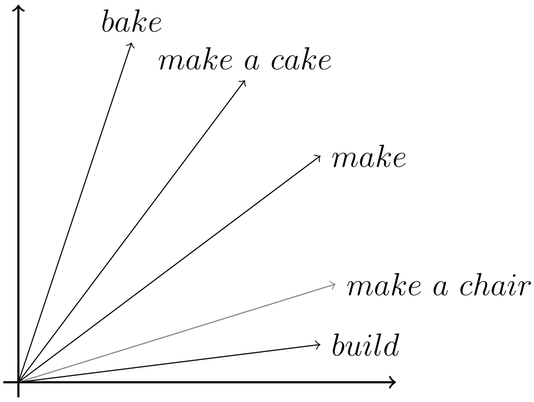

Figure 5. Example of a contextualized vector

4.3 Experiments

This section details our experiments on using distributional methods for computing additional classification features.

For this section of experiments we use the Asp-Ambig corpus of Friedrich and Palmer (2014a) as detailed in Section 3.2.2. Unless otherwise specified, these experiments use Thater et al.'s (2011) distributional model which was trained on the Gigaword corpus, as well as the three seed sets from the LCS Database (Dorr et al., 2001) that were also used by Friedrich and Palmer (2014a). The seed sets contain words from the LCS database (Dorr et al., 2001). These words were extracted from the data base and assigned to a seed set according to their entries in either the dynamic or stative category or in both.

4.3.1 Different Ways of Computing Vector Similarities

The goal of this experiment is to improve the classification of aspectual class by contextualizing the verb instance vectors with their subjects and direct objects before comparing them to the seed set. We experiment with computing the similarity between the contextualized verb vector $verb_c$ and the seed set vectors by obtaining the seed vectors using different approaches.

First, for each of the seed sets we compare the contextualized verb vector to each of the seed word vectors *seedw* and compute their similarities using cosine similarity. Afterwards, the average similarity for each seed set is computed.

$$sim(verb_c, seedset) = \frac{sim(verb_c, seedw_1) + sim(verb_c, seedw_2) + ...sim(verb_c, seedw_n)}{n}$$

The other approach is computing a centroid vector for each seed set and then later computing the similarity between the contextualized verb vector and each seed sets centroid vector.

$$v(seedset_{centroid}) = v(seedw_1) + v(seedw_2) + ... + v(seedw_n)$$

$$sim(verb_c, seedset) = sim(verb_c, seedset_{centroid})$$

We evaluate the system on the Asp-Ambig data set. The evaluation is performed for each verb separately using a Leave-One-Out cross validation with a Random Forest Classifier. We use the same features (linguistic indicator, instance-based, distributional features) as Friedrich and Palmer (2014a) but switch out the distributional features for our features.

The accuracy scores of the systems as well as the percentage of majority class for each verb can be seen in Table 13. The model *acl2014* is Friedrich and Palmer's (2014a) automatic classification system. *v_add-avg* and *v_add-cent* are our classification systems in which the verb and complement vectors have been simply added together. The first system uses the computed average similarity and the second system uses the similarity between the computed centroid vectors. The best performance for each verb is marked in bold, whereas an increased performance of our system over acl2014 is marked in italics and bold.

The evaluation shows that most of these verbs have too strong of a majority class for any significant improvement. For a number of verbs that have a lower majority class, we demonstrate a somewhat systematic improvement using contextualized vectors. The verbs *find, follow, consider, fill, bear*, and *allow* show improvement over both the majority class baseline and Friedrich and Palmer's (2014a) classification model. v_add-avg outperforms the acl20014-model for 9 of the 20 verbs. Overall, v_add-avg performs best with an average accuracy of 72.8%. v_add-cent shows an accuracy of 72.3%.

What kind of context works best for which verbs will be examined in the next experiment.

4.3.2 Contextualizing Vectors

The last experiment shows that adding context improves the classifier accuracy for some verbs. For this experiment we want to further investigate this idea by separately adding either the subject or direct object as context. We believe that at least some verbs will profit from this contextualization and want to investigate which context is best for which verb.

We conduct two separate experiments: one using the verb and its subject and one using the direct object.

VERB	# OF INST.	MAJ CLASS	acl2014	v_add-avg	v_add-cent
feel	132	**96.2**	95.5	95.5	95.5
say	138	**94.9**	92.8	***94.2***	***94.2***
make	137	**92.0**	92.0	92.0	92.0
come	133	88.0	**88.7**	88.0	88.0
take	138	**85.5**	85.5	85.5	85.5
meet	135	83.0	**85.9**	85.9	84.4
stand	137	81.0	**83.2**	81.8	81.0
find	137	74.5	71.5	**76.6**	**75.9**
accept	135	**70.4**	65.9	***67.4***	***68.9***
carry	136	55.9	**63.2**	60.3	56.6
look	138	55.8	**67.4**	65.9	66.7
hold	135	55.6	54.1	***57.8***	**59.3**
show	138	52.9	**66.7**	65.9	65.9
cover	129	52.7	**59.7**	55.0	57.4
appear	136	52.2	**58.1**	56.6	55.9
follow	123	52.0	59.3	**61.8**	***60.2***
consider	138	50.7	62.3	**67.4**	***66.7***
fill	134	47.8	67.2	**69.4**	67.2
bear	136	47.1	75.7	**77.2**	75.0
allow	135	41.5	46.7	**51.1**	***49.6***
macro-avg	2700	66.5	72.1	**72.8**	***72.3***

Table 13. Accuracy for Classification using Number of Computed Vector Similarities

For the subject experiment, we only use verb instances from Asp-Ambig that have a subject. We contextualize the verb vector with its subject during initializing according to Thater et al. (2011). We then compare this verb vector to either the centroid vector of each seed set or to each seed set word individually and calculate the average. The comparison is done by using a number of similarity measures for both cosine similarity and scalar product similarity:

avg This value is computed by computing the average of the similarity values of the instance and each seed set's seed words.

cent This value is computed by computing the similarity between the instance and each seed set's centroid vector.

max This value is computed by picking the highest similarity value between the instance and each seed set's seed words.

3max This value is computed by taking the three highest similarity scores and computing their average.

We include the computed similarity values for each instance as features in our Asp-Ambig data set. We then evaluate the new features by classifying on Friedrich and Palmer's (2014a) linguistic indicator and instance-based features and each of our newly computed distributional features separately.

The results can be seen in Table 14. Although the classifiers with the new features perform better than the majority class on average, Friedrich and Palmer's (2014a) system outperforms them with an average accuracy of 86.3. Going verb by verb, the new features perform better for *accept, show, consider* and *cover*. It seems that for these verbs the subject adds a lot to the disambiguation.

We then perform the same experiment but include the object instead of the subject. Results can be seen in Table 15. The classifiers using contextualized vectors perform better on average with a maximum accuracy of 85.5 by using the centroid cosine similarity measure than Friedrich and Palmer's (2014a) acl-2014 model. It is not surprising that adding the context of the objects is more helpful for classification than adding the subject. Most verb actions are only ever be performed by a person whereas their objects can show great variance. Adding the object to the verb vectors improves classification accuracy for the following verbs: *fill, allow, bear, carry, consider* and *hold*. Mostly, it seems to hold that the less dominant the majority class is, the greater the increase in accuracy when adding more information. This did not hold for *look* which can be explained by instances of *look* not having direct objects but objects occurring in at-PPs or like-PPs which we did not consider.

This evaluation shows that adding context to the verb vectors can be beneficial for classification. Some verbs do better without it, while others need it for disambiguation. Which context should be added depends on the verb, especially for lower majority class verbs. The verbs *accept, show, consider, cover* benefit from contextualization with their subject, while *fill, allow, bear, carry, consider, hold* benefit from their object. Adding context to verbs should therefore be considered on a verb by verb basis.

4.3.3 Different Seed Sets

We also want to re-evaluate the LCS seed sets used by Friedrich and Palmer (2014a). For this part of the experiments we compile multiple other seed sets which we then evaluate for a number of different settings.

We gather a number of seed sets for the states *dynamic, stative* and *both.* The number of seed words for each seed set can be found in Table 16.

LCS The original seed sets used by Friedrich and Palmer (2014a).

LCS-opp-v These seed sets were computed by recalculating the centroid vectors for the LCS seed sets for *dynamic* and *stative.* This was achieved by subtracting the $stative_{centroid}$ vector from the $dynamic_{centroid}$ vector and vice versa:

$$v_{dyn-opp} = v_{dyn} - v_{stat}$$

$$v_{stat-opp} = v_{stat} - v_{dyn}$$

Siegel These seed sets come from an annotated list of verb instances by Siegel and McKeown (2000). If all verb instances were only annotated as *stative*, the verb was added to the *stative* seed set. If they were only annotated as not stative, the verb was added to the *dynamic* seed set. If a verb's instances were annotated as *both*, the verb was added to the *both* seed set.

LCS-Siegel This is a combination of the seed words from LCS and Siegel.

	# OF	MAJ			Cosine Similarity				Scalar product			
VERB	INST	CLASS	acl2014	Add-feat	avg	cent	max	3max	avg	cent	max	3max
make	64	100.0	**100.0**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
meet	80	100.0	**100.0**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
say	121	100.0	**100.0**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
feel	113	99.1	**99.1**	95.6	92.9	94.7	93.8	94.7	94.7	94.7	93.8	0.0
come	106	97.2	**97.2**	96.2	95.3	95.3	96.2	95.3	96.2	96.2	95.3	95.3
accept	61	96.7	96.7	95.1	95.1	95.1	95.1	95.1	**98.4**	**98.4**	95.1	96.7
fill	40	92.5	92.5	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**
bear	24	91.7	**91.7**	83.3	83.3	83.3	83.3	83.3	79.2	79.2	79.2	79.2
follow	39	**89.7**	84.6	84.6	82.1	82.1	82.1	82.1	82.1	82.1	76.9	79.5
find	85	**89.4**	85.9	82.4	83.5	83.5	82.4	83.5	83.5	83.5	82.4	81.2
take	106	88.7	88.7	**91.5**	**91.5**	90.6	**91.5**	**91.5**	90.6	90.6	90.6	90.6
stand	48	87.5	**93.8**	80.0	83.3	77.1	83.3	79.2	81.2	81.2	77.1	79.2
allow	47	74.5	85.1	**89.4**	76.6	76.6	80.9	76.6	78.7	78.7	83.0	83.0
show	74	74.3	**85.1**	74.3	78.4	77.0	67.6	70.3	78.4	78.4	74.3	73.0
hold	44	63.6	61.4	70.5	70.5	70.5	70.5	70.5	**75.0**	**75.0**	70.5	70.5
look	95	61.1	82.1	76.8	70.5	72.6	75.8	76.8	71.6	71.6	75.8	75.8
consider	66	60.6	66.7	63.6	72.7	71.2	68.2	66.7	68.2	68.2	66.7	66.7
appear	110	57.3	69.1	61.8	60.0	59.1	60.0	60.0	60.0	60.0	60.0	60.0
carry	56	57.1	67.9	**75.0**	**75.0**	**75.0**	**75.0**	**75.0**	73.2	73.2	73.2	73.2
cover	36	50.0	77.8	86.1	**88.9**	**88.9**	86.1	86.1	**88.9**	**88.9**	**88.9**	**88.9**
AVG	-	81.6	**86.3**	85.1	84.7	84.4	84.3	84.1	84.7	84.7	83.9	79.4

Table 14. Classification Accuracy for Subject Only Contextualization of Vectors

					Cosine Similarity				Scalar product			
VERB	# OF INST	Maj class	acl2014-model	Add-feat	avg	cent	max	3max	avg	cent	max	3max
make	62	100.0	**100.0**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
meet	64	100.0	**100.0**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
say	14	100.0	**100.0**	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
feel	43	97.7	**97.7**	97.7	97.7	97.7	97.7	97.7	97.7	97.7	97.7	97.7
accept	73	95.9	**95.9**	93.2	93.2	94.5	93.2	93.2	93.2	93.2	93.2	93.2
find	65	95.4	**95.4**	95.4	95.4	95.4	95.4	95.4	95.4	95.4	95.4	95.4
fill	43	95.3	95.3	95.3	97.7	97.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
come	15	93.3	**93.3**	93.3	93.3	93.3	93.3	93.3	93.3	93.3	93.3	93.3
stand	11	**90.9**	83.0	81.8	81.8	81.8	81.8	81.8	81.8	81.8	81.8	81.8
take	107	90.7	**91.6**	86.9	88.8	89.7	87.9	88.8	86.9	86.9	87.9	88.8
follow	30	90.0	**90.0**	70.0	80.0	86.7	76.7	73.3	80.0	80.0	73.3	73.3
allow	59	79.7	**86.4**	79.7	79.7	79.7	81.4	78.0	81.4	81.4	78.0	78.0
bear	29	79.3	82.8	89.7	89.7	89.7	89.7	89.7	**93.1**	**93.1**	89.7	89.7
appear	3	**66.7**	0.0	0.0	**66.7**	**66.7**	0.0	33.3	**66.7**	**66.7**	**66.7**	**66.7**
show	47	66.0	83.0	**85.1**	**85.1**	**85.1**	**85.1**	**85.1**	**85.1**	**85.1**	**85.1**	**85.1**
carry	71	64.8	74.6	70.4	70.4	70.4	70.4	71.8	**76.1**	**76.1**	70.4	73.2
consider	51	58.8	76.5	76.5	72.5	72.5	76.5	74.5	78.4	78.4	80.4	**82.4**
cover	52	57.7	76.9	**82.7**	75.0	75.0	80.8	**82.7**	75.0	75.0	76.9	78.8
hold	40	55.0	65.0	75.0	**80.0**	**80.0**	67.5	72.5	75.0	75.0	75.0	75.0
look	11	**54.5**	36.4	36.4	**54.5**	**54.5**	**54.5**	45.5	36.4	36.4	45.5	45.5
AVG		81.6	81.2	80.5	85.1	**85.5**	81.6	82.8	84.8	84.8	84.5	84.9

Table 15. Classification Accuracy for Object Only Contextualization of Vectors

Seed set	LCS	Siegel	LCS-Siegel	MCS	LCS-MCS
dynamic	3695	39	3721	111	3720
stative	240	256	492	148	287
both	334	27	354	7	337
TOTAL	4269	322	4567	266	4344

Table 16. Number of Seed Words per Seed Set

MCS These seed sets were manually compiled by scouring the web for websites[2] which specify verb examples for the aspectual classes *dynamic, stative* and *both*.

MCS-Siegel This is a combination of the seed words from LCS and MCS.

We also experiment with constructing seed sets using **WordNet** synonyms for the LCS seed words. Unfortunately, this only produces large noisy seed sets with many verbs being added not to one but every set.

For each seed set, we calculate new context features using some of the similarity measurements from before, which we calculated with both cosine and scalar product similarity:

avg This value is the average of the similarities between the instance vector and the seed set word vectors.

cen This value is the similarity between the instance vector and the centroid vector from each seed set.

For each seed set and each feature setting we perform a classification using Leave-One-Out (LOO) and RandomForest. This leaves us with 4 new features x 5 seed sets = 20 configurations. We decide to construct a series of new classifiers by using the classification results of the aforementioned classification settings:

Majority Vote The class that is predicted most often by the classifiers.

[2]Website versions: Jan. 11th 2016

http://web2.uvcs.uvic.ca/elc/studyzone/410/grammar/stat.htm

http://grammar.ccc.commnet.edu/grammar/progressive.htm

http://www.really-learn-english.com/dynamic-verbs-and-stative-verbs.html

http://www.studyenglishtoday.net/stative-dynamic-verbs.html

	# of Accuracy Results better/worse than LCS				
	LCS-opp	LCS-Siegel	Siegel	MCS	LCS-MCS
Maj. Vote	1/11	1/7	5/7	4/9	5/8
Prob. Avg.	4/6	3/6	2/8	6/8	5/5
Prob. Max	6/6	3/4	4/5	7/3	4/6
Prediction Features	-	5/7	6/4	4/5	6/5

Table 17. Summarized Classification Results for Different Seed Sets

Probability Average For every classification the end probability values for each aspectual class were gathered. The average probability for each aspectual class was computed and the class with the highest average probability was chosen.

Probability Max The class that has the highest probability of all the classifier values.

A summarized version of the classification results can be seen in Table 17. More detailed accuracy results can be seen in the Tables 27 - 30 in the Appendix section, as mentioned later in the text.

The detailed evaluation results for the different seed sets while using majority vote can be seen in Table 27. The evaluation shows no improvement over the LCS seed set. Although some of the other seed sets lead to improvements for a number of verbs, on average the original LCS seed set still performs best. Mixing the seed sets also does not lead to any improvements.
The detailed evaluation results for the different seed sets while using probability average can be seen in Table 28. Using this kind of classifier leads to a minor improvement in average accuracy of 0.1 of LCS-MCS over LCS. For most verbs (12/20), the LCS seed set still performs best of all the seed sets.
The evaluation results for the different seed sets using probability max can be seen in Table 29. With this classifier we see an improvement in average accuracy of 0.3 over LCS by MCS. Here, LCS only has the best or equal performance for 8 of 20 verbs.

These experiments do not yield consistent improvement. Therefore, we perform a last experiment of stacking classifiers. This is done by utilizing the predictions from the already performed experiments. For each seed set we take the prediction for each verb instance from the Majority Vote, Probability Average and Probability Maximum classifiers. We include these predictions as features to our test data. Again, we use LOO with a RandomForest classifier. This yields some improvement (see detailed Table 30 in Appendix) compared to the performances of the combined classifiers. LCS and MCS seed set show an average accuracy of 74.4, compared to 70.4 of the LCS-Siegel seed set and 74.3 of Siegel and LCS-MCS.

	Maj. Class	LCS	LCS-Siegel	Siegel	MCS	LCS-MCS
Without LingInd	66.5	72.5	72.3	72.2	72.6	**73.0**
With LingInd	66.5	**72.8**	**72.5**	**72.6**	72.6	72.6

Table 18. Classification Accuracy for Classification with and without Linguistic Indicator Features

To better evaluate our new seed sets we also examined how much Friedrich and Palmer's (2014a) LingInd features actually contribute to the classification. For our experiment we will classify only with the instance-based features and the distributional features using the different seed sets. These results should help explain the accuracy results of the earlier experiments. We conducted this series of classification experiments using LOO and RandomForest classifiers with majority vote. The results in terms of average accuracy can be seen in Table 18 and the detailed results in the Appendix in Table 31. One can see that LingInd do not actually contribute much to the classification on most cases. On average, LingInd features add 0.2 - 0.4 % in accuracy. In one case, it seems like the use of distributional features with a seed set (LCS-MCS) actually worsened accuracy results.

4.4 Summary

In this chapter, we detail our experiments using distributional methods.

We experiment with different methods of computing vector similarities. We show improvement using the average similarity between the verb instance and the seed words for verbs with low majority classes.

We also examine the idea of introducing more context into the classification by including the subject or object in the verb instance vectors. We find no consistent method of computation but show that a number of verbs benefit from adding either the subject or the object to their vectors.

We experiment with choosing a different seed set for Friedrich and Palmer's (2014a) distributional features as well as different classification methods such as combined classifiers and stacking. This does not lead to any consistent improvements over LCS with more than 0.3%. We conclude therefore that this seed set can not be improved and is the best choice for classification. By also classifying with and without LingInd features, we deduce that these only add up to 0.4 % accuracy on average.

5 Improving Classification of Light-Verb-Constructions Using a Rule-based Approach

In this chapter, we address the problem of finding a better handling of so called light-verb-constructions (LVCs). We investigate whether we need a different approach for aspectual class classification of LVCs.

First, we will expand on the problem of classifying LVCs; and also mention related work. After introducing our newly established corpora, we will detail our experiments, including a corpus study, direct object comparisons and finally, a rule based approach.

5.1 Problem

One problem for identifying the aspectual class of verbs are cases where the verb type changes its class depending on the context. Friedrich and Palmer (2014a) hypothesize that a majority of these cases are light verbs.

Light verbs have no set definition. They are Multi-Word-Expressions (MWE) as they always come with complements, the verbs themselves are "light" in the sense of meaning. Therefore they need their complements in order to form real meaning. LVCs usually consist of a verb and an NP (nominal phrase) which is derived from a verb (Kearns, 2002). Some examples can be seen in (1).

(1) take a peek
take a drive
have a cry
have a rest
have a lie-down

LVCs are semi-productive (Vincze et al., 2011b). New constructions following a certain pattern can enter a language at any time (2). Therefore simply keeping a list of LVCs is impossible.

(2) give a call
give a Skype call

Furthermore, not every MWE is an LVC. Unlike literal expressions or idioms, LVCs are semi-compositional. Examples (3) - (5) show three different kinds of MWE using a V-NP construction with the verb "take". The meaning of (3) is literal and can be interpreted by combining the constituents. (4) is an idiom. This means it can not be interpreted literally or as a combination of its constituents. Example (5) is, finally, an LVC. Its meaning can be found by interpreting the verb's complement "walk".

(3)	take a book	Literal
(4)	take a while	Idiom
(5)	take a walk	LVC

Interpreting an MWE is an enormous problem for a range of NLP applications. The same verb and similar complement NP can actually be two very different things. Example (6) is interpreted in a literal sense. In contrast, example (7) is an LVC with the NP carrying most if not all of the meaning.

(7)	The patient had a look of pure mania.	Literal
(8)	Yes, sure. I will have a look.	LVC

A big indicator that a phrase is an LVC is if it can be paraphrased with a single verb without losing any of its meaning (Hwang et al., 2010). Since the verb does not have much meaning, the NP complement can usually be converted to a verb. This can be seen in example (9).

(9)	take a walk	walk
	have a look	look

In the English language (and in many others) verbs in context can be classified as belonging to one class: stative or dynamic. Stative verbs refer to unchanging states or conditions whereas dynamic verbs describe activities or events.

The classification of LVCs is difficult because their lexical aspect depends on the light verb's arguments (10).

(10)	She had a look of pure joy on her face.	stative
	The English man is having a cup of tea.	dynamic

Although both sentences use the same "verb", they represent different lexical aspects which can be regarded as coarse-grained word senses. It is the arguments that determine the final lexical aspect of the verb. This is why handling these LVCs poses such a challenge for NLP applications.

5.2 Related Work

Hwang et al. (2010) introduce an approach for annotating multilingual LVCs in PropBank. Identifying and labelling the arguments of a sentence's predicate can be important for tasks in NLP such as semantic role labelling. One major obstacle one will come across during this task are multiword expressions like LVCs.

The authors tackle this problem by creating frame files for the light verbs they manually annotated in PropBank. These files specify the argument structure as well as syntactic and semantic behaviour of the verb. Their final annotation scheme uses just one frame file per verb. This frame file is a combination of the light verb's and the true predicate's semantics since these can differ from each other.

There have also been numerous approaches on how to identify light-verb-constructions:

Vincze et al. (2011b) investigate the differences between detecting noun compounds and light verbs. The authors test various methods to find an approach that works for both types and can be extended to other kinds of MWEs. The investigated methods include the following: Dictionary-based ones try to match the expression with a dictionary containing noun compounds. A POS-matching rule that marks expressions if they fit certain patterns. Additionally, they check if the noun ends in certain suffixes, if the verb is in a list of the 15 most frequent verbs or if one of the nouns is derived from a verbal stem. Lastly, the authors use methods that look at syntactic relations. All of these methods are tested on a corpus consisting of 50 Wikipedia articles.
The authors find no method that performs consistently for both noun compounds and light verbs. The POS-tagging and dictionary-based methods work best for noun compounds while the most frequent word and noun features, POS-tagging and syntactic information work best for light verbs.

Another approach makes use of multiple languages for automatically identifying LVCs (Vincze et al., 2013). The authors investigate the Hungarian and English language with the use of a parallel corpus and an additional corpus for each language. They then develop six feature sets comprised of a variety of morphological, semantic, orthographical, statistical, lexical and syntactic features. These features are either language-specific or language-independent.
The authors argue that studying topologically different languages is the key to improving current systems which are mostly based on the English language. Therefore, Hungarian, as a morphologically rich language, is investigated. The authors want to prove that features from one language can benefit identification in the other language, thereby leading to innovations for both systems.
The gathered features are then used in a machine learning approach for identifying LVC based on decision trees. The experiments show that all features contribute to the performance and that the features sets turn out to be equally successful for both languages. The F-Score of the English system reaches 0.59, whereas the Hungarian one reaches 0.56 compared to the F-score of the baseline of 0.42 and 0.45 respectively.

5.3 Corpus Data and Statistics

This section details our three compiled corpora. Each corpus consists of sentences for the following six light verbs: *get, give, have, hold, make, take.*

Verb	Dyn	Stat	ALL
get	369	32	401
give	364	32	396
have	4	94	98
hold	200	92	292
make	415	26	441
take	325	6	331
ALL	1677	282	1959

Table 19. Brown-lv1: Aspectual Class Distribution

Verb	Maj. Class	Fleiss' κ	Cohen's κ a_1, a_2	a_1, a_3	a_2, a_3
get	92.0	0.69	0.96	0.98	0.98
give	91.9	0.25	0.87	0.93	0.83
have	95.9	0.45	0.97	0.97	0.98
hold	68.5	0.23	0.51	0.81	0.67
make	94.1	0.38	0.88	0.98	0.89
take	98.2	0.16	0.91	0.99	0.92

Table 20. Brown-lv1: Fleiss' and Cohen's κ

5.3.1 Brown-lv1

The Brown-lv1 corpus consists of 1959 sentences from the Brown Corpus[3]. The Brown Corpus is a commonly used American English corpus with a wide array of genres. The genres include topics such as Press, Religion, Humor and many more. The corpus contains approximately 1 million words. The sample used for this thesis was taken from the NLTK toolkit.

The distribution of sentences can be seen in Table 19. The annotator agreement between the 3 annotators is shown in terms of Fleiss' and Cohen's κ in Table 20. For this corpus only the sentences annotated by the annotators as dynamic or stative were used.

The expected agreement for Cohen's κ was computed by estimating the prior category distribution (Artstein and Poesio, 2008) of our annotators' annotations:

$$P_e = \frac{1}{4i^2} \sum_{k \epsilon K} n_k^2$$

Here, i denotes the total number of items, k a category and n_k the total number of annotations of both annotators to the category k.

5.3.2 Brown-Penn-Wiki50-lv1

The Brown-Penn-Wiki50-lv1 corpus consists of manually annotated sentences from several corpora: Brown, Penn Treebank, and Wiki50.

The Penn Treebank Corpus (Marcus et al., 1993) is a corpus of POS-tagged, parsed and partially annotated texts from various categories. The corpus contains approximately 7

[3]NLTK Version: http://www.nltk.org/data.html

Verb	Brown Corpus	Penn Treebank	Wiki50	ALL
get	403	82	83	568
give	400	65	102	567
have	100	232	100	432
hold	299	40	54	393
make	445	161	170	776
take	332	121	150	603
ALL	1979	701	659	3339

Table 21. Brown-Penn-Wiki50-lv1: Number of Annotated Sentences per Corpus

million words. The sample used for this work is the sample provided by the NLTK toolkit which contains articles of the Wall Street Journal. The Wiki50 Corpus is a small corpus constructed by Vincze et al. (2011a). It contains 50 Wikipedia articles and is manually annotated for multiword expressions and named entities.

For the Brown part, Brown-lv1 and additional sentences were used. The total number of annotated sentences can be found in Table 21. This corpus consists of sentences annotated as dynamic, stative and both.

Annotator agreement is not available for this corpus as it was divided between different annotators as well as partially annotated by only one annotator.

5.3.3 Wiki-lv2

Wiki-lv2 consists of sentences from various Wikipedia articles across genres.

The annotations were done by three annotators. The annotators had to decide on one of four labels (dynamic, stative, both, context). The label "context" is used for labelling the sentences as problematic or when more context is needed for annotating. Out of each annotator's annotation for an instance, the label with the majority vote is picked. Only the sentences with the gold label "dynamic" or "stative" are chosen for testing. Thus, of the 600 annotated (100 per verb) sentences, 520 are chosen in the end (see Table 22 for distribution).

The inter-annotator agreement can be seen in Table 23 for Fleiss' and Cohen's κ. Cohen's κ was computed according to Artstein and Poesio (2008). We report substantial to almost perfect agreement for Cohen's κ.

Verb	Dyn	Stat	ALL
get	95	0	95
give	85	9	94
have	1	46	47
hold	68	30	98
make	75	15	90
take	91	5	96
ALL	415	105	520

Table 22. Wiki-lv2: Aspectual Class Distribution

			Cohen's κ		
Verb	Maj. Class	Fleiss' κ	a_1, a_2	a_1, a_3	a_2, a_3
get	100.00	-0.03	1.00	0.95	0.93
give	90.40	0.40	0.90	0.94	0.90
have	97.90	0.34	0.90	0.95	0.93
hold	69.40	0.56	0.80	0.77	0.86
make	83.30	0.40	0.80	0.86	0.79
take	94.80	0.44	0.98	0.92	0.92

Table 23. Wiki-lv2: Fleiss' and Cohen's κ

5.4 Corpus Study

To better understand LVC in context we examine the following six light verbs which are among the most commonly used light verbs:

get, give, have, hold, make, take

We conduct a corpus study by manually annotating Brown-Penn-Wiki50-lv1. In contrast to our expectations before conducting this annotation study, the majority class for most of these verbs is quite dominant. It is hard to outperform the majority class as a baseline with any kind of classifier. There are only a few cases where the aspectual class of a verb instance differs from the majority class.

5.4.1 Qualitative Analysis

In the case of *make*, most instances are used in a dynamic sense of an object being produced/caused (see examples (1)-(3)). Whenever *make* is used in a stative sense, it seems to mostly refer to situations where an object consists of something (see examples (4)-(5)) or used in a "this makes sense" way.

(1)	"He also makes fun of, and teases, his older sister ."	Dynamic	Wiki50
(2)	"Bailey Controls, based in Wickliffe, Ohio makes computerized industrial controls systems."	Dynamic	Penn Treebank
(3)	"I had felt the draft they were making while mounting the stairs."	Dynamic	Brown Corpus
(4)	"The SEC documents describe those chips, which are made of gallium arsenide, [...] ."	Stative	Penn Treebank
(5)	"Soy sauce made from human hair"	Stative	Wiki50

The majority of *get* instances are of the dynamic class (example (6)) with a small number of instances where *get* is used in the simple past tense in a "have got"-construction (example (7)).

(6)	"Pat, get out of that creek ! !"	Dynamic	Brown Corpus
(7)	"But because it 's all we've got, I'm going to vote for it"	Stative	Penn Treebank

Give is almost exclusively used in a dynamic sense (example (8)) with "on a given day" constructions (example (9)) being the majority of stative instances.

(8)	"Caldwell gave no details, according to Kyle."	Dynamic	Wiki50
(9)	"[...] one manner of experience will be typical of any given group [...] ."	Stative	Brown Corpus

The other verbs show similar tendencies. *Take* is usually used in a dynamic sense with only a few instances being stative. The stative instances seem to mostly be related to duration, such as "take one hour"-kind of constructions.

(10)	"We may take her with us to California."	Dynamic	Brown Corpus
(11)	"[...] the experimental chamber took a long time to refit [...] ."	Stative	Wiki50

Have is the only light verb in this selected group whose majority class is stative. It is mostly used in a sense of owning something (see example (12)). Have can be dynamic when used in an expression such as "having a cup of coffee" (also: see example (13)).

(12)	"Jennifer has a younger brother who is Jimmy 's age."	Stative	Wiki50
(13)	"He had a heart attack."	Dynamic	-

Hold seems to be the only verb in this selection that does not have too strong of a majority class. In its dynamic sense, it is mostly used in physically hold something in your hands-constructions (example (14)), whereas in a stative sense, it is used in a figurative way.

(14)	"Holding the pistol concealed, he walked to the rear wall of the stockade."	Dynamic	Brown Corpus
(15)	"Mrs. Caldwell, a registered nurse, held a degree from Touro Infirmary of New Orleans."	Stative	Wiki50

The analysis shows that there simply are not enough instances of the verbs of the non-majority class. Also, it is difficult to improve upon a system that classifies verbs based upon their majority class when that system already has an accuracy of above 90%.

5.5 Comparison of Direct Objects

The verb instances of *make* in examples (1) and (2) are used in a dynamic sense whereas in example (3) it is used in a stative sense.

(1)	make a *bread*	dynamic
(2)	make a *cake*	dynamic
(3)	make *sense*	stative

If one takes a look at the direct objects (highlighted in italics) one can see that there is a larger similarity between the first two objects, bread and cake, than between any other object combination.

This leads to the hypothesis that it might be possible to predict the aspectual class of an instance by comparing its direct object to other direct objects that occurred with that verb.

Say the instance in question is *make a bread*. One can then compare that instance's direct objects of labelled instances to all other direct objects used with that verb and compute some kind of similarity between them (for example WordNet Similarity). Example (4) demonstrates this (and uses actual WordNet similarities).

(4)	bread	cake	dynamic	2.5
	bread	sense	stative	1.6
	bread	chair	dynamic	1.3
	bread	meal	dynamic	2.3

Once these similarities are computed, the object pair with the highest similarity is chosen and the aspectual class of the second object assigned to the unlabelled instance in question (here: bread - cake: 2.5, dynamic).

We test our approach on a part of Brown-lv1 which consist of 1384 sentences annotated with a direct object. These experiments do not lead to any major improvements due to the strong majority classes of the verbs (see Table 24). During the experiments two kinds of similarity are used: WordNet similarity between the two objects and cosine similarity between the object vectors using Thater et al.'s (2011) VSM model. context+lingInd refers to Friedrich and Palmer's (2014a) approach.

5.6 Rule-based Classification

The corpus study in section 5.4 shows that for most of the investigated light verbs their non-majority class usages are few and conventionalized. This leads to the hypothesis that their classification could be done with a rule-based system.

	% maj class	majority class	context+lingInd	WordNet Sim	VSM Sim
get	91.6	47.1	**96.1**	53.6	46.8
give	91.0	49.3	49.2	**63.5**	49.0
have	94.0	**48.9**	**48.9**	48.6	48.6
hold	66.2	41.9	49.3	**52.1**	44.9
make	93.0	49.3	**74.5**	49.3	49.3
take	97.9	**49.5**	**49.5**	48.5	49.1

Table 24. Classification F1 values Using Comparison of Direct Objects

5.6.1 Development

The majority class of the six light verbs is quite strong. Therefore we develop rules that would match the few cases of the respective minority class. This means we for the verb have we develop rules which will target the dynamic instances, whereas the rules for the other verbs will target their stative usages. For developing rules we use Brown-lv1 as a training corpus. Just like in the corpus study in chapter 5.4, the uses of each verb in the minority class are conventionalized and few. They will be reviewed shortly:

Get often occurs in stative sentences such as (1) and (2). We write rules matching various *has + get* combinations. If a rule matches we check that it is not a verbal construction such as in (3) by applying another regular expression on a lemmatized version of the sentence. We use the NLTK[4] WordNet lemmatizer for lemmatization.

(1)	And he *hasn't* even *got* a knife on him.	Brown
(2)	She*'s got* plenty of guts, Mr. Paxton.	Brown
(3)	She *has gotten checked* for Malaria.	-

Give occurs in a stative sense as in the following examples. Our rule matches sentences containing *given* as an adjectival modifier.

(4)	There was no word spoken, no apparent *signal given.*	Brown
(5)	Hence , if what is in question is whether in a *given theology myth* [...]	Brown

Have often occurs in dynamic sentences such as (6) and (7) with *have*'s object being something food or group activity related. We try developing rules by parsing the sentences and checking for the object belonging to *have*. We then use WordNet and check if any of the object's synsets had a lexical filename of either noun.food or noun.group. This unfortunately does not work well and produces many incorrectly matched sentences. Due to the wide array of dynamic *have* sentences we find it impossible to develop any rules.

[4]Natural Language Toolkit: http://www.nltk.org

(6)	Earlier this month Edward R. Murrow [...] came to Hollywood and *had dinner* [...]	Brown
(7)	[...] he *had* a small *audience* of small children right on stage with him.	Brown

Hold often occurs in stative sentences expressing opinion (examples (8) and (9)). We also explore rules matching *hold + abstract object/physical object* via WordNet hypernyms (see example (10)). Unfortunately this did not result in an improvement.

(8)	The High Court *held that* the company must apply its percentage allowance [...]	Brown
(9)	[...] when you ignore all who *hold* a different *opinion*.	Brown
(10)	Old *attitudes* are *held* more tenaciously in the Tidewater than the Piedmont.	Brown

Make often occurs in stative sentences such as (11) and (12) where it describes a thing's matter or that something is plausible.

(11)	And the monastic communities were supposed to be *made up of* volunteers [...]	Brown
(12)	It is not a mess you can *make sense* of.	Brown

Take often occurs in stative sentences describing an action's duration.

(13)	His looting of the orderly room had *taken* only a *minute* or two [...]	Brown
(14)	it *took* them over an *hour* to get back to the station [...]	Brown

The regular expressions for *hold*, *make* and *take* search a version of the string with lemmatized words. The sentences are first tokenized, then POS-tagged and finally lemmatized using NLTK tokenizer, POS-tagger and lemmatizer respectively. To match the regular expressions for *get* and *give* the sentences are first parsed with the Stanford Parser and the expressions are chosen to match its output.

After a few iterations on the training data, we settle on the final patterns which can be found in Table 25.

5.6.2 Evaluation

To evaluate these rules we used Wiki-lv2 as a test corpus. The performance of the rules can be seen in Table 26. Our rules lead to improvements over the majority class for all but one verb. The test data for that verb, *get*, unfortunately only consists of dynamic sentences and our rules for *get* match stative sentences. Nevertheless, our rule does not match any dynamic sentences.

<table>
<tr><th>Verb</th><th>Regular Expressions</th><th>Matching aspect</th><th>Matching sentence example</th></tr>
<tr><td>hold</td><td>(hold ((\w)+){1,3}opinion)</td><td>Stative</td><td>He held the opinion that...</td></tr>
<tr><td></td><td>(hold that)</td><td>Stative</td><td>The state holds that all...</td></tr>
<tr><td>make</td><td>(make (from|of|up))</td><td>Stative</td><td>Made from cotton.</td></tr>
<tr><td></td><td>(make (\w*){0,3}sense)</td><td>Stative</td><td>That makes sense.</td></tr>
<tr><td>take</td><td>(take (\w*){0,4}(minute|hour|time|day|month|year))</td><td>Stative</td><td>It takes 15 minutes.</td></tr>
<tr><td>give</td><td>(amod\([a-z]+-[0-9]+, given-[0-9]+\)</td><td>Stative</td><td>In the given system...</td></tr>
<tr><td>get</td><td>(aux\(got-(\d)+, has-(\d)+\))</td><td>Stative</td><td>He has got a car.</td></tr>
<tr><td></td><td>(aux\(got-(\d)+, 've-(\d)+\))</td><td>Stative</td><td>I've got four kids.</td></tr>
<tr><td></td><td>(auxpass\(got-(\d)+, 's-(\d)+\))</td><td>Stative</td><td>She's got money.</td></tr>
<tr><td></td><td>(neg\(ai-(\d)+, n't-(\d)+\)\, dep\(ai-(\d)+, got-(\d)+\))</td><td>Stative</td><td>We ain't got nothing.</td></tr>
<tr><td></td><td>(aux\(got-(\d)+, have-(\d)+\))</td><td>Stative</td><td>They have got food.</td></tr>
<tr><td></td><td>Should not match: (have/v get/v [a-z]+/v)</td><td>Stative</td><td>I have gotten checked for...</td></tr>
</table>

Table 25. Rules for LVC classification

5.6.3 Error Analysis

The evaluation for *give* shows some errors with rules matching incorrect sentences (example (1)) as well as some unseen sentences (example (2)).

(1) So in 1894 she had taught classes sometimes weekly and *given* talks noted in newspapers [...]

(2) *Given* these features common with both types of 510, Katie Woodencloak is classified as 510A [...]

Analysing the classification errors for *hold* shows some annotation errors (see example (3) which was tagged as *dynamic* instead of *stative*) but most commonly a variety of unseen sentences ((4) and (5)).

(3) Thinkers such as Rousseau have argued that language originated from emotions while others like Kant have *held* that it originated from rational and logical thought.

(4) Thomas sold the remaining land he *held* in Kentucky in 1814

(5) [...] and the first African American to *hold* the office.

The error analysis for *make* shows annotation errors as well (example (6)) and also wrong classification of "making of" noun constructs. Cases like example (8) are not conveyed by the present set of rules.

(6) The bicameral Congress, *made* up of the Senate and the House of Representatives [...]

(7) The *making* of an 8-minute cartoon short

(8) The ability to capture the essence of nature *makes* the Japanese gardens distinctive and appealing to observers.

The unmatched sentences for *take* only consist of unseen instances such as these examples:

(9) [...] that modern musicians *take* for granted.

(10) Film production can therefore *take* as little as one person [...]

We would have liked to further refine our rule-based classification by additional iterations in training and testing of our rule-based system but this was unfortunately beyond the scope of this thesis.

5.7 Summary

This chapter details our approach to classifying the aspectual class of LVCs. Light-verb-constructions consist of a verb and its complements. Since the verb is light in meaning,

	Majority Class		Rule-based System	
Verb	Accuracy	F-Score	Accuracy	F-Score
get	100.0	100.0	100.0	100.0
give	90.4	47.5	**93.6**	**79.5**
have	**97.9**	**49.5**	-	-
hold	69.4	41.0	**74.5**	**58.4**
make	83.3	45.5	**84.4**	**66.3**
take	94.8	48.7	**96.9**	**77.8**

Table 26. Classification Accuracy of a Rule-based Classification System for LVC

it receives its meaning from these complements. This in turn leads to difficulties during classification, as the lexical aspect also depends on the complements.

The chapter then references some related work on LVCs, such as annotating and identifying them; and describes our created corpora.

We present a corpus study on six light verbs by manually annotating the aforementioned corpora. We report commonly used cases for each verb and show a strong majority class for all of them. During first experiments, we then compare the direct objects of each light verb's instances to each other and classify them according to the highest similarity score. This does not lead to any improvements over the majority class baseline.

The conventionalized usages of the six light verbs lead us to a rule-based classification approach. We detail 12 regular expressions for a total of five of the light verbs and show improvement of up to 5.1% in accuracy compared to the majority class.

6 Conclusion

This section summarizes our work and and details research ideas for future work.

6.1 Results

We examine the automatic classification of a verb's aspectual class. The most basic definition of aspectual class is whether a verb is used in a dynamic or stative sense. Such a classification is important for classifying situation entities, as well as classifying discourse modes.

This thesis introduces previous approaches for the classification of aspectual class, such as Siegel and McKeown's (2000); or Friedrich and Palmer's (2014a) model, an aspectual class classifier that uses linguistically motivated and distributional features for a state-of-the-art classification.

We conduct multiple experiments to construct an improved classifier:

For our first series of experiments, our goal is to improve already existing linguistic indicator features by clustering. We cluster the extracted instance-based linguistic indicator (LingInd) features based on the instance's distributional data. For this we use k-means clustering with a set number of clusters for each verb. We then group the LingInd features by their instance's cluster and re-calculated one set of LingInd features per cluster per verb. For the evaluation, we assign each test instance to one of the clusters and use that cluster's features. This does not lead to any improvements over the majority class or the general linguistic indicators per verb.

The second area of improvement concerns distributional features. We experiment with different ways of computing the vector similarities between the verb instance and the seed sets and show improvements by using the computed average of the vector similarities. We contextualize the vectors for the verb instances with either the subject or direct object of the instance and can report that including certain context leads to improvements for a handful of verbs. The verbs *accept, show, consider, cover* benefit from contextualization with their subject while *fill, allow, bear, carry, consider, hold* benefit from adding the object. Furthermore, we compile a number of new seed sets and compare them to the original one used by Friedrich and Palmer (2014a). We try out different combinations of meta classifiers and experiment set-ups and show some improvement but ultimately can not improve upon the original seed set in a consistent way.

Finally, we investigate the handling of light-verb-constructions (LVCs). We conduct a corpus study, detailing commonly used light verbs and their usages. Our first experiment classifies verb instances based on the instance's direct object's similarity to other instances' direct objects. This does not lead to any improvements, but leads us to the idea of using a rule-based classification approach for LVC. We construct a number of regular expressions which capture the majority of conventionalized uses for five light verbs. Our evaluation shows up to 4.1% improvement over the majority class in terms of accuracy.

6.2 Future Work

In the future, we want to experiment more generally with the features for classification. Some features could be redundant; other feature combinations might lead to better classification accuracy. This could easily be investigated by a series of classification experiments excluding one feature after the other.

The clustering of the linguistic indicators could also be improved in further experiments. Another clustering algorithm might be better suited. A smaller number of clusters might also be better and not divide up the features too much. We would also like to investigate the process of data reduction further. A higher number of dimensions with more retained data might lead to a better classification.

When contextualizing the verb vectors we show that different degrees of context are beneficial to a number of verbs. We would like to start a series of new experiments based on this finding. It would be interesting to find out if verbs could be pre-grouped by what kind of context is beneficial to them without having to perform the actual classification experiments. For the aspectual class classification, we would like to set up a system that uses the best contextual setting for each verb. We believe this would further improve the accuracy of classifying a verb's aspectual class.

We would also like to further investigate our rule-based classification approach by doing more iterations for rule development and testing. We believe the approach can be improved even more by adding new rules or even by combining this approach with Friedrich and Palmer's (2014a) classifier by classifying the instances that are not matched by our rules.

7 Appendix

VERB	Maj class	LCS	LCS-opp-v	LCS-Siegel	Siegel	MCS	LCS-MCS
accept	70.4	70.4	70.4	70.4	69.6	70.4	69.6
allow	41.5	50.4	49.6	50.4	48.1	49.6	48.1
appear	52.2	58.8	55.1	56.6	**59.6**	58.8	**60.3**
bear	47.1	75.7	75.7	75.7	75.0	75.0	**76.5**
carry	55.9	55.9	55.9	**57.4**	**56.6**	**58.8**	**56.6**
come	88.0	88.0	87.2	88.0	88.0	88.0	88.0
consider	50.7	68.8	65.9	67.4	66.7	65.9	68.1
cover	52.7	58.9	57.4	58.1	58.1	56.6	57.4
feel	96.2	95.5	95.5	95.5	95.5	95.5	95.5
fill	47.8	67.2	63.4	67.2	**67.9**	**69.4**	**68.7**
find	74.5	76.6	**77.4**	76.6	75.9	**77.4**	76.6
follow	52.0	57.7	54.5	56.9	56.9	56.9	**58.5**
hold	55.6	58.5	57.0	57.0	58.5	57.0	57.8
look	55.8	67.4	65.9	66.7	67.4	**68.8**	66.7
make	92.0	92.0	92.0	92.0	92.0	92.0	92.0
meet	83.0	86.7	85.9	86.7	**87.4**	85.9	85.2
say	94.2	94.2	94.2	94.2	94.2	94.2	94.2
show	52.9	66.7	65.9	65.2	66.7	64.5	65.9
stand	81.0	81.8	81.8	81.8	81.8	81.8	81.8
take	85.5	85.5	85.5	85.5	**86.2**	85.5	85.5
AVG	66.5	**72.8**	71.8	72.5	72.6	72.6	72.6

Table 27. Classification Accuracy for Different Seed Sets Using Majority Vote

VERB	Maj class	LCS	LCS-opp-v	LCS-Siegel	Siegel	MCS	LCS-MCS
accept	70.4	70.4	70.4	70.4	69.6	70.4	70.4
allow	41.5	48.9	**51.9**	**50.4**	**49.6**	**50.4**	**49.6**
appear	52.2	57.4	**59.6**	55.9	57.4	55.9	**60.3**
bear	47.1	77.2	75.7	77.2	75.7	75.7	76.5
carry	55.9	56.6	56.6	**57.4**	56.6	**57.4**	56.6
come	88.0	88.0	88.0	88.0	88.0	88.0	88.0
consider	50.7	68.1	65.2	67.4	67.4	67.4	68.1
cover	52.7	58.9	57.4	56.6	58.1	58.1	57.4
feel	96.2	95.5	95.5	95.5	95.5	95.5	95.5
fill	47.8	67.9	65.7	66.4	67.2	67.2	67.9
find	74.5	75.9	**77.4**	75.9	75.2	**77.4**	**76.6**
follow	52.0	58.5	56.1	56.9	57.7	57.7	**59.3**
hold	55.6	57.0	57.0	**57.8**	**57.8**	**58.5**	**57.8**
look	55.8	67.4	67.4	67.4	67.4	**68.1**	66.7
make	92.0	92.0	92.0	92.0	92.0	92.0	92.0
meet	83.0	86.7	85.9	86.7	85.9	85.9	85.9
say	94.9	94.2	94.2	94.2	94.2	94.2	94.2
show	52.9	66.7	**67.4**	65.9	66.7	65.9	65.2
stand	81.0	81.8	81.8	81.8	81.8	81.8	81.8
take	85.5	85.5	85.5	85.5	85.5	**86.2**	85.5
AVG	66.5	72.7	72.5	72.5	72.5	72.7	**72.8**

Table 28. Classification Accuracy for Different Seed Sets Using Probability Average

VERB	Maj class	LCS	LCS-opp-v	LCS-Siegel	Siegel	MCS	LCS-MCS
accept	70.4	69.6	69.6	69.6	**70.4**	**70.4**	69.6
allow	41.5	50.4	**52.6**	50.4	50.4	48.9	49.6
appear	52.2	57.4	**58.8**	55.9	56.6	55.9	**58.1**
bear	47.1	76.5	75.7	76.5	75.7	75.7	75.0
carry	55.9	55.9	**57.4**	**58.1**	55.9	**58.8**	55.1
come	88.0	88.0	88.0	88.0	88.0	88.0	88.0
consider	50.7	67.4	65.2	67.4	67.4	67.4	65.9
cover	52.7	56.6	**58.1**	**57.4**	**58.9**	**59.7**	**57.4**
feel	96.2	95.5	95.5	95.5	95.5	95.5	95.5
fill	47.8	67.9	64.9	65.7	67.2	67.9	67.9
find	74.5	77.4	**78.1**	75.9	74.5	77.4	76.6
follow	52.0	58.5	56.1	57.7	**59.3**	**59.3**	**59.3**
hold	55.6	57.0	56.3	57.0	57.0	**57.8**	**57.8**
look	55.8	67.4	67.4	67.4	67.4	**68.1**	67.4
make	92.0	92.0	92.0	92.0	92.0	92.0	92.0
meet	83.0	86.7	85.9	86.7	85.9	86.7	85.9
say	94.9	94.2	94.2	94.2	94.2	94.2	94.2
show	52.9	65.9	**66.7**	**66.7**	**66.7**	65.9	65.2
stand	81.0	81.8	81.8	81.8	81.8	81.8	81.8
take	85.5	85.5	85.5	85.5	85.5	**86.2**	85.5
AVG	66.5	72.6	72.5	72.5	72.5	**72.9**	72.4

Table 29. Classification Accuracy for Different Seed Sets Using Probability Maximum

VERB	Maj Class	LCS	LCS-Siegel	Siegel	MCS	LCS-MCS
accept	70.4	71.9	71.1	71.9	71.9	71.1
allow	41.5	56.3	**57.8**	**57.0**	56.3	**57.0**
appear	52.2	58.1	**60.3**	**61.0**	**61.8**	**62.5**
bear	47.1	80.1	80.1	78.7	78.7	78.7
carry	55.9	63.2	63.2	**64.0**	**64.0**	**64.7**
come	88.0	87.2	87.2	87.2	87.2	**88.0**
consider	50.7	68.1	**69.6**	**71.0**	**68.8**	**68.8**
cover	52.7	57.4	54.3	55.0	**58.1**	55.8
feel	96.2	95.5	95.5	95.5	95.5	95.5
fill	47.8	72.4	67.9	67.9	70.9	70.1
find	74.5	76.6	75.9	76.6	75.9	76.6
follow	52.0	65.0	61.0	63.4	63.4	60.2
hold	55.6	58.5	**60.0**	58.5	58.5	57.8
look	55.8	70.3	**71.7**	70.3	70.3	70.3
make	92.0	92.0	92.0	92.0	92.0	92.0
meet	83.0	85.2	84.4	85.2	84.4	85.2
say	94.9	94.2	94.2	94.2	94.2	94.2
show	52.9	66.7	64.5	**68.1**	66.7	**68.8**
stand	81.0	83.2	83.2	83.2	83.2	83.2
take	85.5	85.5	85.5	**86.2**	85.5	85.5
AVG	66.5	**74.4**	74.0	74.3	**74.4**	74.3

Table 30. Classification Accuracy for Different Seed Sets Using Predictions as Included Features

	Maj	With LingInd					Without LingInd				
VERB	class	LCS	LCS-Siegel	Siegel	MCS	LCS-MCS	LCS	LCS-Siegel	Siegel	MCS	LCS-MCS
accept	70.4	70.4	70.4	69.6	70.4	69.6	69.6	68.9	70.4	69.6	69.6
allow	41.5	50.4	50.4	48.1	49.6	48.1	48.9	48.9	47.4	50.4	50.4
appear	52.2	58.8	56.6	59.6	58.8	60.3	60.3	56.6	55.9	56.6	59.6
bear	47.1	75.7	75.7	75.0	75.0	76.5	76.5	77.2	76.5	76.5	77.2
carry	55.9	55.9	57.4	56.6	58.8	56.6	52.9	54.4	53.7	54.4	55.1
come	88.0	88.0	88.0	88.0	88.0	88.0	88.0	88.0	88.0	88.0	88.0
consider	50.7	68.8	67.4	66.7	65.9	68.1	67.4	65.9	68.1	65.9	67.4
cover	52.7	58.9	58.1	58.1	56.6	57.4	56.6	57.4	57.4	58.1	56.6
feel	96.2	95.5	95.5	95.5	95.5	95.5	95.5	95.5	95.5	95.5	95.5
fill	47.8	67.2	67.2	67.9	69.4	68.7	66.4	66.4	67.2	65.7	67.9
find	74.5	76.6	76.6	75.9	77.4	76.6	76.6	77.4	77.4	77.4	77.4
follow	52.0	57.7	56.9	56.9	56.9	58.5	62.6	61.8	58.5	62.6	61.8
hold	55.6	58.5	57.0	58.5	57.0	57.8	55.6	57.0	57.0	58.5	57.0
look	55.8	67.4	66.7	67.4	68.8	66.7	66.7	65.9	65.9	68.1	68.1
make	92.0	92.0	92.0	92.0	92.0	92.0	92.0	92.0	92.0	92.0	92.0
meet	83.0	86.7	86.7	87.4	85.9	85.2	86.7	87.4	86.7	86.7	86.7
say	94.9	94.2	94.2	94.2	94.2	94.2	94.2	94.2	94.2	94.2	94.2
show	52.9	66.7	65.2	66.7	64.5	65.9	65.9	64.5	65.2	63.8	66.7
stand	81.0	81.8	81.8	81.8	81.8	81.8	81.8	81.8	81.8	81.8	81.8
take	85.5	85.5	85.5	86.2	85.5	85.5	85.5	85.5	85.5	85.5	86.2
AVG	66.5	72.8	72.5	72.6	72.6	72.6	72.5	72.3	72.2	72.6	73.0

Table 31. Classification Accuracy for Classification with and without Linguistic Indicator Features

References

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Bach, E. (1986). The algebra of events. *Linguistics and philosophy*, 9(1):5–16.

Baker, K. (2005). Singular value decomposition tutorial. *The Ohio State University*, 2005:1–24.

Béjar Alonso, J. et al. (2013). K-means vs mini batch k-means: a comparison.

Dorr, B. J., Olsen, M., Habash, N., and Thomas, S. (2001). Lcs verb database. *Online Software Database of Lexical*.

Dowty, D. R. (1979). *Word meaning and Montague grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ*, volume 7. Springer.

Friedrich, A. and Palmer, A. (2014a). Automatic prediction of aspectual class of verbs in context. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL). Baltimore, USA.*

Friedrich, A. and Palmer, A. (2014b). Situation entity annotation. *LAW VIII*, page 149.

Graff, D. and Cieri, C. (2003). English gigaword, ldc catalog no. *LDC2003T05. Linguistic Data Consortium, University of Pennsylvania.*

Hwang, J. D., Bhatia, A., Bonial, C., Mansouri, A., Vaidya, A., Xue, N., and Palmer, M. (2010). Propbank annotation of multilingual light verb constructions. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 82–90. Association for Computational Linguistics.

Ide, N., Fellbaum, C., Baker, C., and Passonneau, R. (2010). The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the ACL 2010 conference short papers*, pages 68–73. Association for Computational Linguistics.

Kearns, K. (2002). Light verbs in english.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Moens, M. and Steedman, M. (1988). Temporal ontology and temporal reference. *Computational linguistics*, 14(2):15–28.

Palmer, A., Ponvert, E., Baldridge, J., and Smith, C. (2007). A sequence labeling model for situation entity classification. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 896.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Siegel, E. V. and McKeown, K. R. (2000). Learning methods to combine linguistic indicators: Improving aspectual classification and revealing linguistic insights. *Computational Linguistics*, 26(4):595–628.

Smith, C. S. (2003). *Modes of discourse: The local structure of texts*. Cambridge University Press Cambridge.

Smith, C. S. (2005). Aspectual entities and tense in discourse. In *Aspectual Inquiries*, pages 223–237. Springer.

Thater, S., Fürstenau, H., and Pinkal, M. (2011). Word meaning in context: A simple and effective vector model. In *IJCNLP*, pages 1134–1143.

Vendler, Z. (1967). Linguistics in philosophy.

Vincze, V., Nagy, I., and Berend, G. (2011a). Multiword expressions and named entities in the wiki50 corpus. In *RANLP*, pages 289–295.

Vincze, V., Nagy, I., and Farkas, R. (2013). Identifying english and hungarian light verb constructions: A contrastive approach. In *ACL (2)*, pages 255–261.

Vincze, V., Nagy, T. I., and Berend, G. (2011b). Detecting noun compounds and light verb constructions: a contrastive study. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 116–121. Association for Computational Linguistics.