Linköping University | Department of Computer and Information Science
Bachelor's thesis, 18 ECTS | Cognitive Science
Spring term 2024 | LIU-IDA/KOGVET-G--24/020--SE

Automatic Speech Act Classification

Bootstrapping an embedding-based classifier from a rule-based classifier for Swedish sentences

Daniel Tufvesson

Supervisor: Lars Ahrenberg
Examiner: Arne Jönsson

LINKÖPINGS UNIVERSITET

Abstract

When we speak, we carry out social actions that are mediated through spoken words. For example, through speech, we may *ask* for directions. These spoken actions are referred to as *speech acts.* We humans unconsciously understand and categorize speech acts all the time. But, how can we make computers do the same?

The objective of this thesis was to develop an automatic classifier for speech acts in Swedish sentences. To do this, I first annotated a test set of speech acts, following the MATTER development cycle. The sentences in this test set originate from online discussion forums. I then developed and trained a rule-based classifier using a subsample of these sentences. Finally, this rule-based classifier was used for automatically annotating a large training set, which was then used for training a neural network for classifying speech acts—essentially *bootstrapping* the network from the rule-based classifier. This neural network uses SBERT to compute the sentence embeddings of the sentences and then classifies their speech acts based on these embeddings.

The results indicate that using the MATTER cycle is a feasible approach for creating a test set for speech acts. Furthermore, the results show that the embedding-based classifier outperforms the rule-based classifier, but also that the rule-based classifier vastly outperforms the baseline. However, it was not possible to conclude if the embedding-based classifier's higher performance was due to the increase in data or because of its differing architecture.

Keywords: NLP, Speech Act, SBERT, Rule-Based Classification, Semi-supervised Learning.

Acknowledgement

At first, like is often the case when I do not understand the intricacies of a subject, speech act classification seemed to me like a simple, almost trivial, task. As I started working on it, however, it dawned on me that this task was much harder than I first imagined. I am therefore indebted and grateful to the people who helped me and thereby made this thesis possible.

Thank you to my supervisor Lars Ahrenberg whose guidance and feedback have been invaluable. It is great when one's supervisor is both an expert in the field and also willing to reliably share this knowledge. I would also like to express my gratitude to Arne Jönsson, both for being the examiner of this thesis and for handing me this research topic. At the start, when I had to choose a topic for my thesis, I already knew that I wanted to work with machine learning and natural language processing, but I did not know what specifically. Arne came up with many ideas and suggestions, with speech acts classification as one of them.

My gratitude also extends to Daniel Holmer who provided much-needed technical support for running the neural network code on the Linköping University's computer system. Without his help, my neural network would still be training to this day on my personal laptop. Further, I want to thank my friends Ida Allander, Philip Nguyen, and Louisa Hirvonen for reading my thesis and providing feedback about the writing. I know they were busy with their own theses, and I am grateful for them putting aside some of their valuable time to help me.

All this intellectual support that I have gotten has been immensely important and should not be understated. However, a thesis is not the product of intelligence alone. This thesis would not have been possible without the emotional and social as well as intellectual support I have gotten from my wonderful friend Maria Pusker. Spending almost every day together and working on our respective theses, our routine afternoon walks to Pressbyrån for coffee and fika, and our rants and tirades about the challenges and difficulties we encountered during our work—these have kept my mind sane and energized to undertake this thesis. She has made this entire endeavor so much more fun and enjoyable. Thank you!

Daniel Tufvesson

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

What is done through speaking? How do we use words to influence the world around us? In a sense, a spoken utterance is just a string of vocal sounds. But in another sense, it is also a social action that has real effects on the world. When I say "Pass me the salt", I am not just producing words, I am *requesting* the salt to be handed to me. And assuming the listener is cooperative, they may indeed hand me the salt. In this way, my act of producing vocal sounds is also an act of requesting the salt, and in effect an act of retrieving the salt. These different types of acts are referred to as *speech acts*—what we *do* when speaking.

There are several different speech acts. For example, through speaking we may *complain*, *apologize*, *disagree*, *greet*, *compliment*, *invite*, *inform*, *console*, or *support*, to name only a few. It has been suggested that there is essentially an arbitrary number of speech acts (Yule, 1996). However, while recounting all the possible speech acts may not be a fruitful endeavor, linguists and philosophers have proposed taxonomies—systems of classification—for categorizing speech acts.

In this thesis, I will focus on the speech acts defined in *The Swedish Academy Grammar* (*Svenska Akademiens Grammatik;* Teleman et al., 1999):

- *Assertive* (Sw. påstående): the speaker holds that the content of the sentence is true or at least true to a varying degree. For example: "They launched a car into space."

- *Question* (Sw. fråga): the speaker requests information regarding whether or not something is true, or under what conditions it is true. For example: "Are you busy?" or "How much does the car cost?".

- *Directive* (Sw. uppmaning): the speaker attempts to get the listener to carry out the action described by the sentence. For example: "Open the door!" or "Will you hold this for me?"

- *Expressive* (Sw. värderande inställning): the speaker expresses some feeling or emotional attitude about the content of the sentence. For example: "What an adorable dog!" or "The Avengers are awesome!"

While recognizing the speech act of an utterance may seem like a mundane and simple task, it is so only for us humans. For a computer, the task becomes more difficult. How can we program a computer to recognize speech acts? Or perhaps in more technical terms, how can make a computer automatically classify speech acts of sentences?

1.1 Motivation

A domain where automatic speech act classification could prove useful is corpus linguistics, where large amounts of natural text—a corpus, or corpora in plural—are analyzed to test hypotheses about linguistic concepts (Hunston, 2006). Corpora can also be analyzed to find new patterns of language that would otherwise be difficult to find. Analyses of these kinds often require these corpora to be annotated with linguistic features, for example, parts-of-speech, morphology, syntactic structure, sentiment, or even speech acts.

However, one obvious issue is that natural texts do not come annotated from the start. The annotation work is left to the researcher. Annotating by hand is often a laborious and time-consuming task, so many of these tasks are therefore automated using Natural Language Processing (NLP) and machine learning systems. Currently, most work on these automatic systems has focused on annotating parts-of-speech, morphology, syntactic structure, sentiment, and named entities. Some work has of course been done on automatic speech act classification, but much remains to be done. This is no less true for the Swedish language, as most work has focused on speech acts in the English language.

So, the central question of this thesis is then, how should this be done?

1.2 Aim

The objective of this thesis is to develop an automatic classifier for speech acts that is intended for Swedish sentences written in messages in online discussion forums. The speech acts are the four I have mentioned above: assertive, question, directive, and expressive. I have divided this objective into the following sub-goals:

- Create a test set of annotated sentences for evaluating the performance of speech act classifiers.

- Develop and train a rule-based speech act classifier. This classifier identifies the speech act of a sentence by analyzing its syntactical and grammatical features.

- Bootstrap a neural network to classify speech acts using sentence embeddings. This classifier identifies the speech act of a sentence by analyzing its semantic meaning. The embeddings are computed with a Sentence BERT (SBERT) language model (Reimers & Gurevych, 2019).

In a broader sense, the objective is to contribute to the field of NLP and corpus linguistics, by exploring methods for automatically annotating speech acts.

The classifier models and code are available on GitHub[1] and the data sets are available on Kaggle[2].

[1] https://github.com/Daniel-B-Tufvesson/speech-act-classifier

[2] https://www.kaggle.com/datasets/danieltufvesson/swedics-speech-acts

1.3 Research Questions

The research questions for this thesis are the following:

> **RQ1**: How can we create a test set of sentences annotated with speech acts for evaluating the performance of a classifier?
>
> **RQ2**: How can we use semi-automatic methods for creating a large training set of sentences annotated with speech acts?
>
> **RQ3**: How can a pre-trained SBERT language model be used to classify speech acts of written Swedish sentences?

1.4 Delimitations

While speech acts typically concern spoken utterances, I have instead focused on written messages over the Internet. I highlight this because written and spoken grammar can sometimes differ. What is true for the grammar of a written sentence may not be true for a spoken one. I chose to focus on written messages because such data was more readily available. Furthermore, these sentences are classified and analyzed in isolation from the rest of the dialogues from which they originate.

1.5 Thesis Structure

This thesis is structured into seven chapters (including this chapter).

- Chapter 2: I begin by presenting speech act theory and other relevant linguistic theories.
- Chapter 3: I then describe of the relevant NLP and machine learning technologies, as well as the annotation methodology that supports this thesis.
- Chapter 4: Then I provide a full account of how the data sets were created, how the rule-based classifier works, and how the embedding-based classifier was trained and evaluated.
- Chapter 5: Here I present the data sets and the results from the evaluations.
- Chapter 6: In this chapter, I interpret the results in relation to the research questions, as well as discuss some of the methodological shortcomings of this thesis.
- Chapter 7: Finally, I summarize the main findings and contributions of this thesis.

Chapter 2
Speech Acts and Linguistic Theories

We have seen that a speech act is what we *do* with an utterance. But what makes an utterance a specific speech act and not another? And how do these speech acts grammatically differ from one another? I will in this chapter describe the theory and the taxonomy of speech acts that are central to this thesis. I will also give an account of some of the linguistic features that are relevant to both speech acts and the objective of this thesis.

2.1 Speech Acts

To reiterate, a speech act is a social action that is carried out when the speaker makes an utterance (Yule, 1996). In chapter 1, I used the utterance "Pass me the salt" as an example of a way for the speaker to request the salt from the listener. The speaker can also request the salt by saying "Can you pass the salt?", "Is there any salt?", "This needs more salt", "Minion, I demand the salt!" or just "Salt". In this way the speaker can carry out the same speech act, but by making different utterances.

A speech act can be categorized at different granularities. A speaker may *inform, suggest, insist, refute, explain, conclude,* or *deduce*, which are all examples of specific speech acts. However, in doing these, the speaker is also *asserting*, that is, stating that what is being said is true or false. Asserting is here an example of a general speech act. Asserting can be said to be a *type* of speech act since it covers a wide range of specific speech acts. What type a specific speech act belongs to depends on the taxonomy: the set of rules and criteria for classifying speech acts.

A well-established taxonomy is the one developed by Searle (1969; 1979), which is based on the taxonomy by Austin (1975). Searle's taxonomy consists of five speech acts: *assertives*, *directives*, *commissives*, *expressives*, and *declarations*.

The Swedish Academy Grammar Taxonomy

Teleman et al. (1999) define a similar yet different taxonomy of speech acts. I have already presented most of these in chapter 1, but I will describe them here in greater detail. Some of the types correspond to Searle's (1979), while others do not. Note that when translating these labels from Swedish to English, I have borrowed some of the vocabulary from Searle.

- *Assertives* (Sw. påstående): the speaker holds that something is true or false (or somewhere in between). For example, "They launched a car into space" or "That is not what I meant". The speaker may also express the degree of commitment, that is, how certain the speaker is that something is true or false. For example, "The sky is definitely blue" or "That may be the case".

- *Directives* (Sw. uppmaning): the speaker attempts to get the listener to carry out a specific action. For example: "Open the door!" or "Will you hold this for me?"

- *Questions* (Sw. fråga): the speaker requests information about whether or not something is true, or under what conditions it is true. For example: “How much does the car cost?” or “Are you busy?”.

- *Expressives* (Sw. värderande inställning): the speaker expresses some feeling or emotional attitude concerning the propositional content. For example: “What an adorable dog!” or “The Avengers are awesome!”

- *Hypothesis* (Sw. hypotes): the speaker states something without holding it as true or false, and without expressing any emotional attitude about it. For example: “Imagine if the multiverse exists.” However, note that this speech act is not included in this thesis.

Central to all these is that they are based on the speaker’s intentions and actions. An assertive expresses the speaker’s belief. A directive expresses the speaker’s desire for the listener to do something. A question expresses the speaker’s desire for information. An expressive expresses the speaker’s feelings. And finally, the hypothesis is put forward by the speaker. Hence, there is a distinction between performing a speech act and reporting the speech act of someone else, for example, "John asked a question".

Direct and Indirect Speech Acts

As we shall see in section 2.2, each of these speech acts is often expressed with a typical syntax. When a speech act is expressed with its typical syntax, it is said to be a *direct speech act*. In contrast, *indirect speech acts* are not expressed with their typical syntax, but rather with the typical syntax of another speech act (Yule, 1996). An often referred to example is the request “Can you pass the salt?”. Here the main clause of the sentence is in interrogative form, which typically denotes a question. However, except for in perhaps the most obscure situations, this sentence is not a request for information regarding the listener’s salt passing competencies. Instead, it is an indirect request given in the form of a question, and therefore in fact a directive.

Written Speech Acts in Social Media

While speech acts pertain primarily to spoken utterances, they also apply to written utterances such as those in messages in computer-mediated communication. Speech acts can be found in emails (Cohen et al., 2004), tweets (Vosoughi & Roy, 2016), status messages (Carr et al., 2012), instant messages (Moldovan et al., 2011), and forums (Chaka, 2020; Joksimović et al., 2020; Arguello & Shaffer, 2015). For these, the primary means of communication is through text messages, and hence many non-verbal cues are restricted, such as tone of voice, gesture, and gaze. This restriction increases the risks of ambiguity and misinterpretation of the messages (Trevino et al., 1987). However, because of these risks, people sometimes try to compensate, and even overcompensate, by making their messages more playful, affectionate, and deep (Walther & Burgoon, 1992). So, speech acts do occur in communication in social media, but the language is in a sense textually richer than that of spoken communication.

2.2 Grammatical Features of Speech Acts

As mentioned in section 2.1, speech acts can sometimes be identified from the grammatical features and structure of the sentence. Concerning the structure, speech acts often correspond to certain clause types. Further, some examples of grammatical features are punctuation, verb form, verb mood, and the presence of a finite verb. I will here describe the features that are relevant to each speech act.

Assertives

According to Teleman et al. (1999), assertives are typically expressed with sentences given by a *declarative* clause. Declarative clauses often have a clause base, followed by a finite verb, the subject, and then the remainder of the sentence. Alternatively, the subject can also be part of the clause base. Sometimes the clause base is omitted, and the sentence starts with the finite verb. Table 2.1 contains some examples. Regarding punctuation, periods (“.”) are typically used for assertive sentences. However, sometimes exclamation marks are used instead.

Table 2.1: The word order of the declarative clause.

Clause base	Finite verb	Subject	Remainder
Han	gick		hem.
På morgonen	sprang	hästen	snabbare.
	Går		dit nu.

Questions

Questions are often expressed with *interrogative* sentences (Teleman et al., 1999). There are two types of interrogative sentences: *rogative* (Sw. rogativ) and *quesitive* (Sw. kvesitiv). A rogative sentence typically expresses a yes or no question. It often lacks a clause base and therefore starts with the finite verb. See Table 2.2 for some examples.

Table 2.2: The word order of the rogative clause. Examples are taken from Teleman et al. (1999).

Finite verb	Subject	Remainder
Har	musiken	förlorat all betydelse?
Skulle	vi	kunna få en vit duk på bordet?
Spelar	någon av herrarna	något instrument?

A quesitive sentence often expresses a wh-question, that is, under what conditions something is true. Quesitive clauses have an interrogative clause base, meaning the clause base contains an *interrogative pronoun* or *interrogative adverb*. Interrogative pronouns specify what thing or things are being asked about, such as “vad”, “vem”, or “vilken”. Interrogative adverbs specify the type of condition that is being asked about, such as “hur”,

“var”, “när”, or “varför”. The interrogative pronoun and adverb are usually the first word in the sentence. See Table 2.3 for some examples.

Table 2.3: The word order of the quesitive clause. Examples are taken from Teleman et al. (1999).

Clause base	Finite verb	Subject	Remainder
Vad	kan	Herbert	ha menat med de orden?
Varför	ska	de	inte få göra det?
Hur	får	jag	ihop femton stycken?

As for punctuation, questions often end with question marks (“?”). This is also the case when questions are expressed with other clause types. For example, the question “He went home?” is expressed with a declarative clause, but the question mark signifies that it is a question.

Directives

Directives are typically expressed with *directive* sentences, that is, sentences where the main clause is a *directive clause* (Teleman et al., 1999). The directive clause starts with the finite verb, which is in imperative verb form. Directive clauses sometimes lack a subject. In cases where it is included, it is always a second-person pronoun. For some examples, see Table 2.4.

Directive sentences can sometimes end with exclamation marks (“!”), but other times with periods (“.”). The use of periods is often when the directive is intended to not be an order or command.

Table 2.4: The word order of the directive clause. Examples are taken from Teleman et al. (1999).

Finite verb	Subject	Remainder
Tag	du	med dig paraplyt.
Lägg		säcken under granen
Försök		klappa hunden.

Expressives

Finally, expressives are typically expressed with sentences where the main clause is an *expressive clause* (Teleman et al., 1999). Expressive clauses can start in three different ways:

1. With the subordinating conjunction “att”: “Att man inte kan vara säker på något!”.
2. With the subordinating conjunction “som”: “Som du ser ut!”.
3. With an expressive clause base: “Vad bra det blev till slut!”

Expressive clause bases contain the expressive markers “så”, “vad”, “sådan”, or “vilken”, and these are often placed first (Teleman et al., 1999). The words can form the expressive clause base on their own, or be part of a phrase that constitutes the base. For example:

- Adverb phrase: “***Så*** *mycket fortare* hon springer nu!”
- Adjective phrase: “***Vad*** *vacker* den är!”
- Noun phrase: “***Så*** *fin kjol* du har sytt!”

Expressives clauses can also start with a noun phrase as a clause base, lacking any of the expressive markers (Teleman et al., 1999). In these cases, the noun phrase contains an adjective. For example:

- “*Väldigt höga granar* här är!”
- “*Fin bil* du har köpt!”
- “*Intressanta bilder* du gör!”

While expressives are many times expressed with expressive clauses, this is far from always the case. Expressives are commonly expressed with declarative sentences as well (Teleman et al., 1999). For example:

- “Den var sämst!”
- “Bilen är väldigt fin!”

Finally, for expressives both periods (“.”) and exclamation marks (“!”) can be used for punctuation (Teleman et al., 1999). A period signifies a more neutral expression of emotion and feeling, while an exclamation mark signifies higher intensity.

2.3 Dependency Relations

To automatically analyze the syntactic structure of a sentence, we need a way to represent the structure. The syntactic structure is essentially how the words in a sentence functionally relate to each other. One way to represent this structure is with *dependency grammars* (Jurafsky & Martin, 2023). Here the structure is represented as a tree, consisting of 1-to-1 relations between the words (see Figure 2.1).

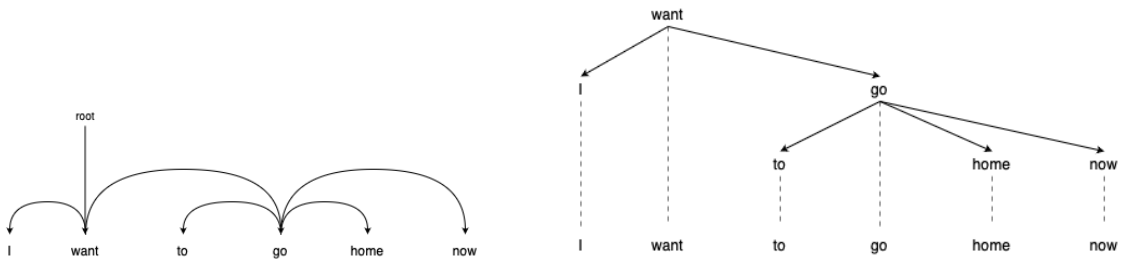


Figure 2.1: The dependency structure of a sentence. The structure is identical in both figures, but are illustrated differently. The left figure also highlights the root word.

A relation constitutes a *head* word and a *dependent* word: head → dependent. The word has only one head but can have several dependents. For example, in Figure 2.1, the word "want" is the head of the dependents "I" and "go". Every word has a head except for the *root* word, which is the head of the entire sentence. For instance, in Figure 2.1, the root word is "want", and thereby constitutes the head of the sentence.

The dependency relations do not only describe the structure but also the grammatical function of the dependents in relation to their heads (Jurafsky & Martin, 2023). A dependent could, for example, function as a subject, a direct object, or an adverbial modifier (see Figure 2.2).

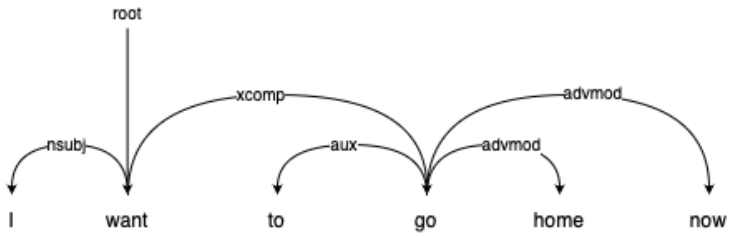


Figure 2.2: Each relation represents a function of the dependent. For instance, "I" is the subject to "want", and "now" is an adverbial modifier to "go".

Universal Dependencies

So, the syntactic structure of a sentence can be represented with dependency relations. How can these relations be represented in a corpora and thereby be made available for computational analysis? The Universal Dependencies framework is a standard for annotating several different linguistic properties across multiple languages. Included in these properties are dependency relations. The framework provides a set of universal tags for their grammatical functions (Universal Dependencies, n.d.-c; De Marneffe et al., 2021). Figure 2.2 consists of some of these tags, and Table 2.5 lists some additional tags. Universal Dependencies hence provide a means for representing the syntactic structure of a sentence and the grammatical functions of its words.

Table 2.5: Some of the Universal Dependencies relations and examples. Bold = dependent. Cursive = head.

Tag	Relation	Example
nsubj	Nominal subject	**Sherlock** *solved* the case.
obj	Direct object	John *drove* the **car** to Baker Street.
iobj	Indirect object	Mycroft *handed* **them** the pictures.
ccomp	Clausal complement	Adler *says* that you **know** the passcode.
det	Determiner	**The** *murderer* is still here.
conj	Conjunct	*Sherlock* and **Moriarty** stared at each other.

2.4 Conversation Analysis and Next-turn Proof Procedure

This thesis concerns itself with identifying speech acts by examining isolated sentences. However, it is worth mentioning alternative methodologies that consider more context. One such methodology is applied in the field of Conversation Analysis. Here it is observed that natural conversation follows a turn-taking system, that is, the participants take turns talking (Sacks et al., 1974). Crucially, in conversation, the participants try to understand each other. A participant's understanding is then displayed when they respond by saying something appropriate in turn. As an example, participant B understands that A's utterance is a question, and therefore produces an appropriate answer. Further, A, in turn, understands that B's utterance is an answer, and displays this by acknowledging it as an answer.

A: What are you doing later?
B: I'm gonna go ice-skating.
A: Cool!

Importantly, the displays of understanding are also available to us, the analysts. We can understand an utterance by how it is responded to. This is referred to as the *next-turn proof procedure.* Thus, to understand the speech act of utterance, we would need to examine the responses. Practically, this means that one would examine an extract from a conversation consisting of several conversational turns, as opposed to examining only an utterance in isolation—indeed a very different approach and objective of this thesis.

Chapter 3

Machine Learning and Natural Language Processing

Central to this thesis is the use of machine learning, in particular, neural networks and rule learning systems. How do these systems classify data? How are they used in NLP? How are they trained from data? And where do data come from? I will here give a brief account of the machine learning technologies and methodologies that are relevant to this thesis.

3.1 Machine Learning

Machine learning systems are computer programs that can predict or estimate values based on a given input data point (Lindholm et al., 2022). These programs are programmed by having them extract patterns and correlations from data. It is in this sense that they are said to learn or train. In general, they can learn two types of tasks: *regression* and *classification*. Regression is the task of approximating a continuous function from discrete numerical data. Classification, on the other hand, is the task of predicting predefined classes from the data (see Figure 3.1). In the case of this thesis, the classes are speech acts, and the data are sentences. This is a case of single-class classification, where only one class is assigned for each data point. Multi-class classification, in contrast, is where multiple classes can be assigned to the same data point.

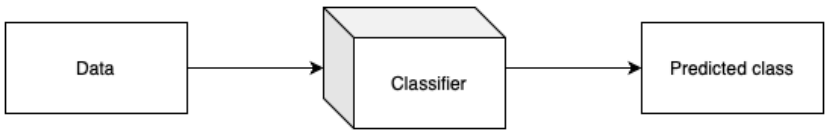


Figure 3.1: The basic operation of a classifier. Data is presented as input, from which the classifier produces a predicted class as output.

Training a Machine Learning System

When training these systems, the data are often divided into three data sets: a *training set*, a *dev set* (or development set; sometimes also called a validation set), and a *test set* (Russell & Norvig, 2022). The training set is used for training the machine learning system. The dev set is used for evaluating the system during training, to determine which hyperparameters to use, such as the type of optimizer, learning rate, and number of trainable parameters. When the training is complete, the test set is used for evaluating the final system to assess how well it performs on unseen data.

Training is done by having a learning algorithm read the data from the training set, and then adjust the system's trainable parameters to reflect the features in the data (Russell &

Norvig, 2022). There are different methods for training a machine learning system depending on the task and the data. Here are a few:

- *Supervised learning*: This method requires the data to be annotated with the expected values to be estimated by the system (Russell & Norvig, 2022). In the case of this thesis, sentences need to be annotated with speech acts. This can be done by manually annotating data with the expected value (see Section 3.12).

- *Unsupervised learning*: This method does not require the data to be annotated. The system learns statistical patterns directly from the data and does not try to associate them with expected values (Russell & Norvig, 2022).

- *Semi-supervised learning*: This method requires only a small amount of the data to be annotated, permitting the rest to be unannotated. From the annotated data, information is extracted and used for annotating the remaining data (Lindholm et al., 2020).

There are further methods for training machine learning systems, for example, self-supervised learning, and reinforcement learning. However, these are not relevant to this thesis.

3.2 Evaluating a Classifier

Once the classifier has been trained, it is important to estimate its performance on unseen data (Russell & Norvig, 2022). We want to know if the classifier has learned the general patterns in the data, or if it has just learned to memorize each data point in the training set—referred to as *overfitting*. Evaluating the performance is done by testing the classifier on the test set. Usually, some single-value metric is computed on how well the classifier performs.

Single-Value Metrics

The most straightforward metric is the classification *accuracy*, which is the percentage of correctly classified data points (Jurafsky & Martin, 2023).

$$\text{accuracy} = \frac{\text{correct classifications}}{\text{total classifications}}$$

Then there is also *precision* and *recall*, which are both class-relative metrics, meaning they are calculated for each class (Jurafsky & Martin, 2023). The precision of a class c is the percentage of data points that are correctly classified as c in relation to all data points classified as c. The recall of a class c is the percentage of data points correctly classified as c in relation to the total number of data points in the data belonging to c.

$$\text{precision}(c) = \frac{\text{correctly classified as } c}{\text{classified as } c}$$

$$\text{recall}(c) = \frac{\text{correctly classified as } c}{\text{total actual } c}$$

Precision and recall hence tell us how well the classifier performs on each class. For example, a classifier may be better at identifying certain classes while being worse at others. Precision and recall can also be combined into a single metric for a class: the *F1-score* (Jurafsky & Martin, 2023).

$$\mathrm{F1}(c) = \frac{2 \cdot \mathrm{precision}(c) \cdot \mathrm{recall}(c)}{\mathrm{precision}(c) + \mathrm{recall}(c)}$$

A decent classifier should strike a balance between precision and recall, and the F1-score gives us a measurement of this balance.

As I have mentioned, precision, recall, and F1 each measure the classifier's performance on each class. This is in contrast to accuracy which measures the overall performance across all classes. It is possible to do the same with the F1-score. This is done by averaging the measure across all classes (Jurafsky & Martin, 2023).

Comparing to a Baseline

These metrics put values on the classifier's performance. However, these values do not tell us much on their own. To be useful, they must be compared to a baseline. One such baseline is the *most frequent class* baseline (Jurafsky & Martin, 2023). This is a classifier that classifies all data points with the most frequent class occurring in the training set. No matter what the specific data point is, it will always be classified with the most frequent class. As an example, assume that c is the most frequent class in the training set and that 84% of the data points in the test set are also of class c. In this case, the baseline accuracy would be 84%. A classifier should therefore have a higher accuracy than this baseline. This also applies to the other measures mentioned above.

In the special case that the data is balanced (see section 3.5), there is no most frequent class, since all classes are equally frequent. The baseline therefore becomes equivalent to classifying purely by chance. So, for two classes, the baseline is 50%; for three it is 33%; for four it is 25%, and so on.

Visualizing the Performance: Confusion Matrices

As we have seen, precision, recall, and F1 can give us indications of how well a classifier performs on each class. However, when working with many classes, this can often lead to an overwhelming amount of numbers to keep track of. Hence, a way to get a quick visual oversight of the class-specific performance is with a *confusion matrix*. In this matrix the rows are the correct classes and the columns are the predicted classes. Each cell in the matrix contains the counts of each test case. For example, in Figure 3.2, the classifier has classified 30 data points as class 1, when in fact they are class 3. A decent classifier should form a noticeable diagonal in the matrix, indicating that is classifying the data correctly. Hence, the confusion matrix visualizes both the classes it performs well on, as well as the ones it does not.

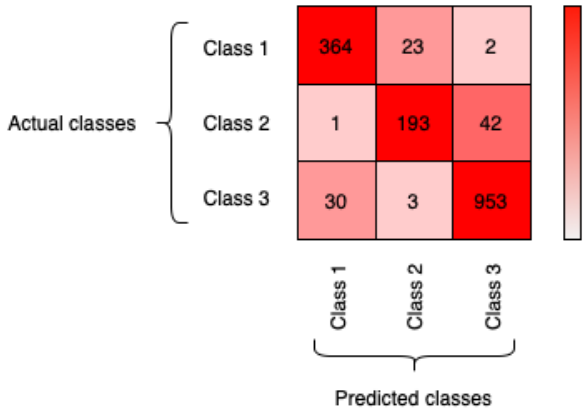


Figure 3.2: A confusion matrix with three classes. Note the diagonal that is formed by the correct classifications.

3.3 Neural Networks for Classification

What I have described so far applies to machine learning systems in general. However, one specific type of machine learning system is the neural network. This type of system was originally inspired by the function of the neurons in the human brain. However, they differ in a great many ways from actual neurons and are therefore best seen as only abstract, idealized, mathematical systems.

An artificial neuron, or *unit*, consists of a set of *weights* and an *activation function* (Russell & Norvig, 2022). The weights determine the strength of each input value x. The weighted inputs are then summed and sent through the activation function. The activation function is nonlinear and produces the output *activation* y of the unit (see Figure 3.3).

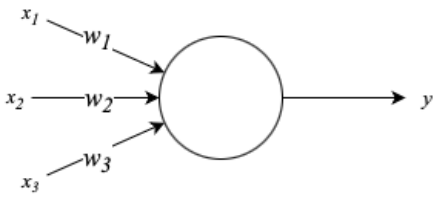


Figure 3.3: A single unit in a neural network.

Units like these can be connected to form neural networks. There are many different types of neural networks, however, the one of interest to this thesis is the *fully connected feed-forward network* (Russell & Norvig, 2022). This network consists of a sequence of layers, where each layer consists of units. Each unit in a layer is connected to each unit in the previous layer (see Figure 3.4). A network consists in general of three types of layers: the input layer, the output layer, and the hidden layers.

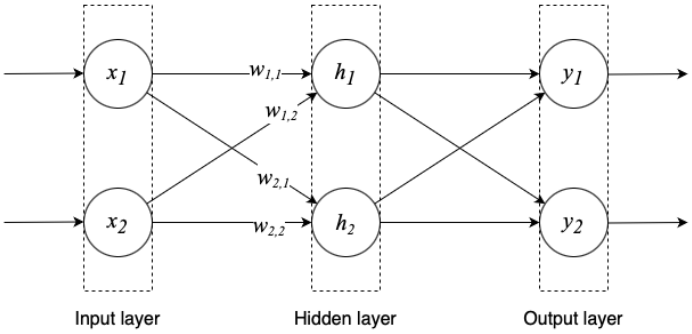


Figure 3.4: A simple neural network that consists of an input layer, a hidden layer, and an output layer. Each layer is made up of two units.

The Input Layer

The input layer is the first layer and is what receives the input data. A data point should be provided as a vector, that is, a list of arithmetic values. For example, $X = (x_1, x_2, \ldots, x_n)$. These values are assigned to each node in the input layer. Hence, the input layer only holds the inputs; it does not transform them in any way.

The Output Layer

The output layer is the final layer and is what produces the output of the network. Similar to the input data, the output is given as a vector $Y = (y_1, y_2, \ldots, y_m)$. Here each value is the value of each node in the output layer. For classification, each output node represents a class. The values are hence the *class scores* for the given data point. The data point is classified with the class that has the highest class score.

The Hidden Layers

In between the input and the output layers, are the hidden layers. These layers connect the input layer to the output layer. The first hidden layer takes the values from the input layer, transforms them based using its weights and activation function, and propagates the transformed values to the next hidden layer. This transformation of data is done in each hidden layer. Finally, the last hidden layer propagates the transformed data to the output layer, which produces the output of the network.

Networks with hidden layers are referred to as *deep networks*. However, it is possible, and sometimes advantageous, to have no hidden layers, and instead only the input layer connected directly to the output layer. These networks are sometimes referred to as *single-layer networks*, because the output layer, in contrast to the input layer, is the only layer that performs any transformations.

3.4 Rule-based Classification

So far I have covered classification with neural networks. The other type of classifier of interest to this thesis is the rule-based classifier. I will here give a rudimentary description of rule-based classification. For an extended description, see Li and Liu (2014).

Rules

In general, a rule is an *if-then* statement, consisting of a *condition* followed by a *conclusion*:

IF condition **THEN** conclusion

If the condition is true, then we can infer the conclusion. For classification, the condition concerns the features of the data, and the conclusion is the predicted class.

IF data point has features X **THEN** predict class y

The classifier holds a set of these rules and tries to find the rule that matches—or *covers*—the given data point. There are different strategies for selecting and learning these rules, two of which are *Rule Induction* and *Association Rule Mining*.

Rule Induction

In Rule Induction systems, the rules are stored in an ordered *decision list*. To classify a data point, the classifier iterates through this list to find the first rule that covers the data point. Once a rule is found, the class is predicted with this rule.

To learn the rules, the classifier begins by generating a rule, checks which data points in the training set this rule covers, and then removes these data points. The rule is then added to the decision list. This process of rule generation and data removal is repeated until all the data points have been removed. Hence, the classifier has learned all the rules that cover the training set.

Class Association Rule Mining

The second strategy is to mine *class association rules*. This strategy finds all the rules with features that are associated with a target class. An example of a rule would be:

IF features X **THEN** predict class y [*support*=10%, *confidence*=80%]

Here, X is a subset of all features that appear in the training data, and Y is the target class. Each rule also has a *support* and *confidence* value.

- The support is the percentage of data points that have the features X and belong to the class y. It tells us how often the rule can be applied in the training set.

- The confidence is the percentage of data points, among those with the features *X,* that belong to the class *y*. It tells us how often the rule can be applied to data points that have the features *X*.

Learning the rules consists of finding all the rules that cover the training set. This is in contrast to rule induction, which only finds some of all the possible rules. In addition, these rules should also satisfy a *minimum support* and a *minimum confidence*.

When classifying, there are different methods for finding the appropriate class based on these rules. One method is to use the strongest rule. Specifically, for a given data point, find the strongest rule that covers the data point, and use that for classification. The strength of a rule can be measured in multiple ways. It can for example be a combination of the rule's confidence and support.

General vs. Specific Strategies

Both Rule Induction and Class Association Rule Mining are general strategies for learning rules and classifying data. However, it is possible to create specific strategies better adjusted for the particular data that is to be classified. One example is XRules which is used for classifying structural data such as XML.

3.5 Dealing with Unbalanced Data

One issue when training machine learning systems for classification is the risk of unbalanced data (Russell & Norvig, 2022). Often data of each class is not equally distributed. If the neural network is trained on a highly imbalanced dataset, then it is at risk of only learning to predict the most frequent class in the data.

One solution is to balance the dataset (Russell & Norvig, 2022). Balancing can be done either through *downsampling* or *upsampling*. Downsampling is when we remove data from all classes that are larger than the smallest class so that all classes become the same size as the smallest. One problem, however, is that we risk losing large amounts of data. The alternative is upsampling, where data are duplicated in all classes that are smaller than the largest class. All classes then become the same size as the largest class. The specific data points to duplicate are selected randomly. Upsampling thus allows us to keep all the data while still having it balanced.

Another solution, in contrast to balancing, is to use *class weights* when calculating the loss (Russell & Norvig, 2022). This means we weight the loss depending on the expected class so that the loss is downscaled for frequent classes, and upscaled for infrequent ones. This essentially has the same effect as balancing the data. In contrast to upsampling, it does not require more storage or a longer training duration.

3.6 Embeddings and Vector Semantics

As mentioned in section 3.3, neural networks rely on vectors as inputs. So, when using neural networks for words and sentences, how do we represent these as vectors? Further, how do we ensure that these vectors encode their semantic meaning? This is the domain of vector

semantics. Words are represented as vectors—*embeddings*—in a high-dimensional semantic space (Jurafsky & Martin, 2023). In this space, vectors that are closer to each other share similar meanings. For example, the vector representations of the words "car" and "truck" are closer to each other than those of "car" and "neutron". This space is derived from how words co-occur with other words. The words "car" and "truck" co-occur in more sentences than "car" and "neutron".

That meaning is derivable from co-occurrence is known as the *distributional hypothesis*, which essentially states that two words have the same meaning if they have the same neighboring words (Jurafsky & Martin, 2023). Some suggest that this is one of the ways we humans learn the meaning of words (Landauer, 2014). We can learn the meaning of a new word by comparing it to other words that would occur in the same context. For example:

> The *nuftply* has two turbofans, a larger wingspan than previous models and can carry up to 150 passengers.

While "nuftply" is not a real word, it would nonetheless be possible to infer that it is some type of aircraft, due to the contextual words "turbofans", "wingspan", "models", and "passengers".

Hence, the meaning of words and sentences can be represented with *word embeddings* and *sentence embeddings* respectively. Further, their meanings can also be compared with other words and sentences.

Computing Word Embeddings

How do we compute these embeddings? There are several techniques available, for example, word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). These models are examples of *static* embedding models, that is, models that produce fixed embeddings for each word, no matter what context they are in (Jurafsky & Martin, 2023). For instance, the word "mouse" has the same embedding whether it is in the context of computer-related words or animal-related words. A *dynamic* or *contextualized* embedding model, in contrast, produces embeddings that take these context words into account. In other words, for these models, the meaning of a word depends on the context as well. One such contextualized model is the *Bidirectional Encoder Representations from Transformers* (BERT) language model (Devlin et al., 2019).

Computing Sentence Embeddings with SBERT

While BERT is suitable for computing contextualized word embeddings, it is less so for sentence embeddings. One approach has been to compute the embeddings of each word in a sentence, and then calculate the average embedding of these, thus producing an embedding that represents the entire sentence. This has however not produced satisfying results on evaluation tasks (Reimers & Gurevych, 2019), essentially failing to capture the semantics of the sentences.

An alternative approach, which has yielded far higher performance, is the *Sentence-BERT* (SBERT) model (Reimers & Gurevych, 2019). SBERT is an extension of BERT, which, in contrast to BERT, directly produces semantically meaningful sentence embeddings. They

are meaningful in the sense described earlier, namely sentences that have similar meanings are closer to each other in the semantic space.

SBERT was originally trained on English sentences (Reimers & Gurevych, 2019), making it unsuitable for Swedish sentences. However, a Swedish SBERT model has since then been developed (Rekathati, 2021).

3.7 Sentiment Analysis

We have seen how expressive speech acts expresses emotions and feelings (see Section 2.1). Emotions often contain some degree of valence or hedonic tone, which is either positive or negative. Within NLP, the expressed valence in a text is referred to as *sentiment*. The task of identifying the sentiment in a text—*sentiment analysis*—is a type of machine learning classification task (Jurafsky & Martin, 2023. Typically, the sentiment is either *positive*, *negative*, or *neutral* (see Figure 3.5).

Positive: "These apples taste great!"

Negative: "I have some awful news."

Neutral: "The book is on the table."

One approach is to use BERT to classify the sentiment of a text. For example, in this thesis, I have used a BERT model which was fine-tuned by Hägglöf (2023) to classify Swedish texts (see section 4.2). This classifier also provides the class scores (see section 3.3) as probabilities for each sentiment (see Figure 3.5). Other techniques are to use a Naive Bayes classifier, LSTM, or sentiment lexicons (Jurafsky & Martin, 2023).

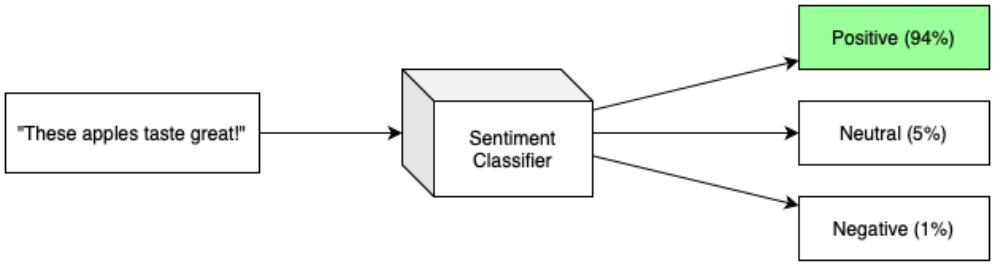


Figure 3.5: A sentence is classified with a positive sentiment with a 94% confidence.

However, one issue with the classification of sentiment is that it rests on the assumption that a sentence expresses a single sentiment (Liu & Zhang, 2012). The problem arises for sentences with mixed sentiment, for example, "The book is great, but the movie is terrible". Should such a sentence be classified as positive, negative, or neutral? It certainly expresses sentiment—in fact, two sentiments—however, none of the classes are suitable. One solution is to use a *mixed* class (Sentiment - Amazon Comprehend, n.d.).

In sum, sentiment analysis can give us a means for automatically assessing the emotional tone in a sentence.

3.8 The Stanza Neural Pipeline

So far I have covered NLP systems from a theoretical perspective. However, when working with these systems in practice, it is convenient to do so within a code framework that can process and store the text data. In this thesis, I have used the Python package Stanza[3] which is one such framework. This package provides tools for segmenting texts into words and sentences, parsing part-of-speech, morphological features, lemmas, dependency parsing, as well as sentiment analysis. Currently, these tools support up to 70 languages, including Swedish. Stanza also supports the Universal Dependencies formalism, meaning it has support for loading and saving linguistic data with the CoNLL-U format.

Stanza processes text through a pipeline of neural models. As input, it accepts raw text or already annotated text, which it then passes through a sequence of neural models, each responsible for performing specific NLP tasks, such as part-of-speech tagging and dependency parsing. What specific models should be included in the pipeline is specified by the user. Figure 3.6 shows an example of what such a pipeline could look like. Stanza thereby provides a means for flexibly processing linguistic data.

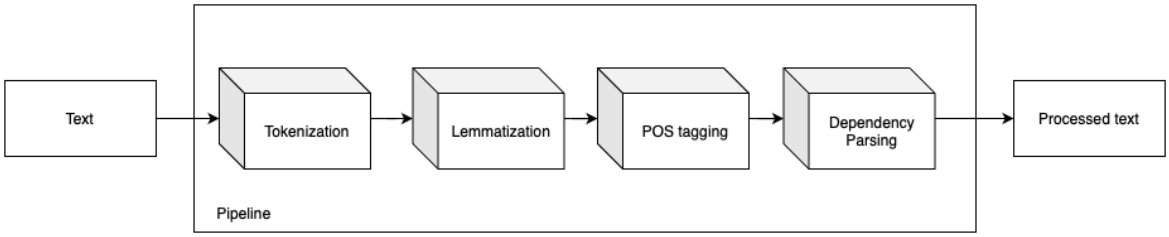


Figure 3.6: In Stanza, text is processed through a pipeline of neural models.

3.9 Automatic Speech Act Classification

We have so far seen how some machine learning systems work and how some of them are applied in NLP. But what about machine learning applied to speech act classification? Most work on automatic speech act classification has been in the context of *dialogue systems*, that is, computer systems that can carry out a dialogue with a human user (Jurafsky & Martin, 2023). In these systems it is not speech acts per se that are being classified, but rather *dialogue acts*. These are similar to speech acts, except that they are classified in relation to the current *dialogue state*. The system classifies the user's dialogue act, and from this generates an appropriate response. For example, if the user asks a question, the system will generate a suitable answer.

However, while the term dialogue act is specific to dialogue systems, it is often used interchangeably with speech acts. Several classifiers and classification methods have been developed, both for dialogue acts and speech acts. Here are a few:

[3] https://stanfordnlp.github.io/stanza

- Mast et al. (1996) developed a dialogue act classifier using semantic classification trees.

- Georgila et al. (2009) developed a computational tool for the automatic annotation of task-oriented dialogue data.

- Stolcke et al. (2000) developed a statistical approach for classifying dialogue acts in natural conversation (as opposed to a dialogue system machine).

- Jeong et al. (2009) developed a semi-supervised method for classifying speech acts in emails and internet forums.

- Suendermann et al. (2009) trained a statistical classifier for dialogue acts. They trained it on dialogues that were automatically annotated by a rule-based classifier. (I will further describe their approach in section 3.11.)

- Saha et al. (2020) fine-tuned a BERT model to classify speech acts in English tweets. It can classify seven different speech acts.

- Vosoughi & Roy (2016) developed a logistic regression classifier for speech acts in tweets. For this, they used both semantic and syntactic features. The semantic features included, for instance, *opinion words*, *vulgar words*, *emoticons*, and *speech act verbs*. The syntactic features included, for instance, *parts-of-speech*, d*ependency sub-trees*, and *punctuation*.

These systems all use different taxonomies of speech acts, where some are more granular, while others are less so. Some are task-specific, and others are generic. What is central to all of these classifiers that I mentioned is that they classify English sentences.

3.10 Speech Act Corpora

Not only have speech act classifiers been developed, but annotated corpora have been as well. Much like the classifiers, the number of speech acts varies greatly for each corpus. Some are more granular, while others consist of only a few speech acts. Here are some of the corpora:

- The DailyDialog corpus (Li et al., 2017) consists of 13,118 manually annotated dialogues. The original dialogues are written by human authors and are intended for second-language English learners to learn English dialogues from. The speech acts are: *Inform*, *Question*, *Directive*, and *Commissive*.

- The Switchboard corpus (Calhoun et al., 2010) consists of 42 different dialogue acts. It is a manually annotated corpus consisting of 1,155 human-to-human phone conversations in English. Some of the dialogue acts are *Statement*, *Opinion*, *Yes-No-Question*, *Yes answers*, *Quotation*, *Hedge*, and *Signal-non-understanding*.

- The SPAAC corpus (Leech & Weisser, 2003) consists of 1,219 English dialogues and 41 different speech acts. The speech act classes are intended to be specific enough to be used in task-oriented dialogues, but generic enough to cover a wide range of dialogues.

So, there are several speech act corpora, and much like the speech act classifiers, they consist of different speech act taxonomies, and have been developed for different purposes in mind.

3.11 Bootstrapping a Classifier

Central to this thesis is the concept of *bootstrapping*. The term should not be confused with the resampling method in statistics which is also termed bootstrapping (Efron, 1979). Instead, bootstrapping is here a looser term, broadly referring to the procedure of automatically creating an improved machine learning system from a more basic system. For example, creating a more accurate classifier from a less accurate classifier.

Semi-supervised training (see section 3.1) is essentially a form of bootstrapping. Here we start with a small set of annotated data and a large set of unannotated data. The annotated data is used for extracting useful features and information, which can then be used for automatically annotating the remaining data. One semi-supervised approach is *self-training* (Amini et al., 2023). Here, a classifier is essentially being trained on its own output (see Figure 3.7). First, the classifier is trained on the annotated data set. Then, a subset of the classifier is used for classifying a subset of the unannotated data. If a classified data point is confident enough, it is added to the training set, which is then used for training the classifier again. This process of classification and training is repeated until all of the unannotated data has been classified. Hence, the classifier is training itself, requiring only a small initial set of annotated data to get started.

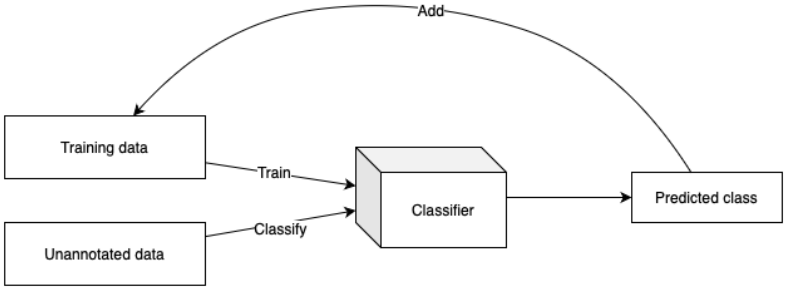


Figure 3.7: Self-training of a classifier. Unannotated data are classified and added to the training set.

Perhaps most relevant to this thesis is the bootstrapping approach used by Suendermann et al. (2009), where they bootstrapped a statistical classifier for dialogue acts from a rule-based classifier (see Figure 3.8). The dialogue acts are specific dialogue systems for phone call routing and caller troubleshooting. The rule-based classifier consisted of handwritten rules for the grammar of the dialogue acts. It was then used for classifying utterances for the training set. The statistical classifier consisted of a trigram language model and a naive Bayes classifier. Comparing their performances, the rule-based classifier had an accuracy of 78%, while the bootstrapped statistical classifier achieved an accuracy of 90%. Hence, it is a viable approach to bootstrap a statistical classifier from a rule-based one.

As we shall see in Chapter 4, this thesis builds on both bootstrapping from rules and the semi-supervised approach of starting with a small set of annotated data.

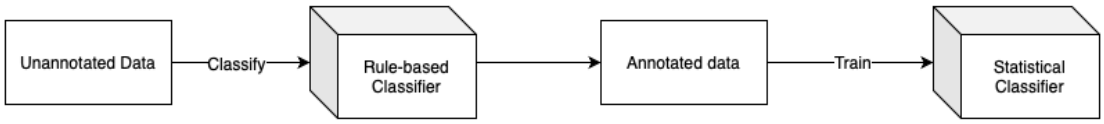


Figure 3.8: Bootstrapping a statistical classifier from a rule-based one.

3.12 Linguistic Annotation

When training and testing a machine learning system, you need annotated data. Where do you get annotated data from? One approach is the bootstrapping method that I mentioned above, which can sometimes work for training data. But what about test data? The test data need to be accurate so that we can be sure that the algorithm is actually classifying data correctly. One straightforward, albeit tedious, approach is to annotate the data manually. This means that a human annotator (or several human annotators) would go through each data point (for example each sentence) in the test data and assign it the correct class label.

At first glance, this may seem like a simple (and almost mindless) task, but there are a few pitfalls, especially regarding scientific *reliability* and *replicability*. The annotations must be reliable, meaning that the annotator will annotate the same sentence with the same class every time. For example, the annotator should always annotate the sentence "This sucks" as an expressive each time it occurs.

Closely related to reliability is that the annotations must also be replicable, meaning that if other annotators later try to carry out the annotation work, they should end up with the same annotated data. They should also end up annotating “This sucks” as an expressive.

The MATTER cycle

To ensure that the annotation work is scientifically rigorous, it is important to have a standardized annotation methodology. One such methodology is the *MATTER* development cycle described by Pustejovsky and Stubbs (2013). This cycle consists of six steps.

1. **Model**: A model of the linguistic phenomenon to be annotated is created from existing linguistic theories. The model consists of the annotation labels and how these labels relate to the data. For example, what speech acts are to be labeled, and what makes a particular sentence belong to a specific speech act and not another? The complexity of the model can vary depending on the linguistic phenomena it intends to cover.

2. **Annotate**: The data is then annotated by human annotators. To their help is a set of annotation guidelines which were created based on the model. The guidelines describe the phenomena, the labels, and the conditions for what label to be annotated to what data. The guidelines may also contain examples, as well as instructions on what to do with ambiguous edge cases.

3. **Train**: A machine learning system is trained on a training set of annotated data.

4. **Test**: The machine learning system is tested on a test set of annotated data.

5. **Evaluate**: The results are evaluated.

6. **Revise**: Based on the evaluation, the model and the annotation guidelines are revised to be better suited for the machine learning system.

These steps are then repeated until the machine learning system and the model are evaluated to be sufficient for their intended purpose.

The MAMA Sub-cycle

Creating an accurate model that fully captures the linguistic features to be annotated is seldom done on the first try. Therefore, the two initial steps, model and annotate, are done iteratively several times. This iterative process is referred to as the Model-Annotate-Model-Annotate sub-cycle (or MAMA sub-cycle) and is a part of the full MATTER cycle (Pustejovsky & Stubbs, 2013). First, an initial model is created based on linguistic theories, from which a set of annotation guidelines are formulated. Then, using the guidelines, a small subsample of the data is annotated. This is done as an initial test of the model and often reveals things that the model does not cover, for example, ambiguous sentences that belong to several classes (or perhaps no class at all). In these cases, the model needs to be revised to cover the problematic data. Another test annotation is then done to uncover more problematic cases. This process is repeated until the revised model is deemed to sufficiently cover the linguistic features to be annotated.

A way to determine if a model is sufficient is to compute the inter-rater agreement between two annotators of the test annotations. A high agreement indicates that the model is sufficient, while a low indicates that the model may need to be revised.

3.13 Inter-rater Agreement

The *inter-rater agreement* is essentially how much two, or several, annotators agree with each other (Pustejovsky & Stubbs, 2013). Then there is also *intra-rater agreement* which is how much a single annotator agrees with himself over repeated annotations.

Cohen's Kappa

This is often measured with a statistical value κ (kappa). One such value is Cohen's kappa, which measures the inter-rater agreement between two annotators (Pustejovsky & Stubbs, 2013). Cohen's kappa takes into account that agreement can occur purely by chance and is hence more accurately representing the agreement than just a percentage.

While this thesis relies only on Cohen's kappa, there are other measures of agreement, such as Fleiss's K (Fleiss, 1971), Cronbach's Alpha (Cronbach, 1951), and Krippendorff's Alpha (Krippendorff, 1970).

Interpreting the Kappa

How is this kappa value interpreted? The interpretation depends on the task and its difficulty. There are, however, a few general guidelines that have been developed for certain tasks. One of these is the scale proposed by Landis and Koch (1977, as cited in Pustejovsky & Stubbs, 2013). The different levels of the scale are listed in Table 3.1. So, by using Cohen's kappa and the agreement scale, we can estimate the reliability of the two annotators and the annotation guidelines they are using.

Table: 3.1: A scale for interpreting the kappa value.

κ	Agreement
0.8–1.0	Perfect
0.6–0.8	Substantial
0.4–0.6	Moderate
0.2–0.4	Fair
0.0–0.2	Slight
< 0.0	Poor

Chapter 4
Implementation

The main objective of this thesis was to develop an automatic classifier for speech acts. This was done by bootstrapping an embedding-based classifier from a rule-based classifier. The bootstrapping was done in the following steps:

1. Collect and pre-process sentence data from Swedish corpora (see Section 4.1).

2. Create a test data set by first developing an annotation tool and then annotating a sample of the data (see Section 4.2).

3. Develop and train a rule-based speech act classifier (see Section 4.3).

4. Create a training data set by automatically annotating the remaining data that was not included in the test data set (see Section 4.4).

5. Train an embedding-based speech act classifier using the automatically annotated training set (see Section 4.5).

6. Evaluate and compare the rule-based and embedding-based classifiers with the manually annotated test data (see Section 4.6).

These individual steps are described in detail below.

4.1 Data Collection and Preprocessing

The data used in this thesis were retrieved from different Swedish corpora from Språkbanken Text[4] and are given in Appendix A. These corpora consist of randomly ordered, annotated sentences. Each sentence consists of tokenized words and linguistic features of each word, such as part-of-speech tags, dependency relations, and morphosyntactic properties. The sentences originate from two internet discussion forums[5]. Each corpus revolves around some discussion topic, such as parenting, cars, politics, or science. I have therefore chosen a large number of corpora in order to maximize the variety of topics, and thereby minimize the effects of topic-specific features on the classification.

Each Språkbanken Text corpus was annotated in XML, but to be efficiently parsed in Stanza, I converted them into CoNLL-U (Universal Dependencies, n.d.-a; see also Buchholz & Marsi, 2006). Further, some of the syntactic tags used in these corpora differ from Universal Dependencies tags which the models in Stanza rely on. The differing tags that were

[4] https://spraakbanken.gu.se

[5] https://www.flashback.org and https://www.familjeliv.se

relevant to this thesis were the part-of-speech tags and the dependency relation tags. I converted the part-of-speech tags directly by following the conversion table given by Universal Dependencies (Universal Dependencies, n.d.-b). For the dependency relation tags, however, there was at the time no direct conversion method available. Instead, using the Swedish dependency relations tagger in Stanza, I retagged all sentences in each corpus. A similar solution was applied by Nivre & Megyesi (2007) for harmonizing two corpora, and they refer to this as annotation projection.

In addition to this, I also automatically tagged each sentence with its sentiment, that is, whether a sentence is positive, negative or neutral (see section 3.7). This was done using a classifier developed by KBLab (Hägglöf, 2023).

This resulted in a data set of 3,800,000 sentences with their tokens and syntactic properties, formatted as CoNLL-U and compatible with Stanza. These were then used for creating the test and training data sets.

4.2 Creating a Test and Dev Data Set

The test data was created by manually annotating a subsample of the collected data. This subsample was created by extracting the first 1,000 sentences from each corpus and then collectively shuffling them. This resulted in a subsample of 38,000 sentences that were to be annotated. Creating a test set out of these involved developing an annotation tool, developing a set of annotation guidelines, annotating the sentences, and finally post-processing and cleaning up the annotated data.

Annotation Labels

The sentences were annotated with the following speech labels: *Assertive*, *Question*, *Directive*, *Expressive*, as well as two additional labels: *None* and *Unsure*. The None label is a garbage class meaning that the sentence does not express any of the given speech acts. The Unsure label was for ambiguous cases not covered by the annotation guidelines. The reason for differentiating the None and Unsure labels, as opposed to just having one label, is because it contributed to developing the guidelines. The None label was covered by the guidelines, while the Unsure was not, thus showing where more development needed to be done.

Developing the Annotation Tool

To aid my annotation work, I developed a simple, browser-based annotation tool. This tool consists of a graphical user interface that displays a sentence to be annotated and buttons with speech act labels (see Figure 4.1). Pressing a button annotates the sentence with that label, and displays the next label. The annotation work is done in sessions of 50 sentences. This is to divide the work into smaller more manageable tasks and to give me as an annotator a sense of progress.

Figure 4.1: The annotation tool that was developed. Clicking a button annotates the current sentence. In case of a misclick, clicking the undo button reverts the previous annotation. The labels: Assertive (Sw. påstående), Question (Sw. fråga), Directive (Sw. uppmaning), Expressive (Sw. expressiv), Unsure (Sw. vet ej), and Other (Sw. annat)

Developing the Annotation Guidelines

I developed a set of annotation guidelines which I then followed during the annotation work (see Appendix B). Annotation guidelines are, according to the MATTER cycle (see section 3.12), drafted from a formal model, which is initially developed from linguistic theories. However, I did not develop an explicit model, but instead developed the guidelines directly from theories about speech acts (see section 2.1) and their grammar (see section 2.2). Then, similar to the MAMA sub-cycle, I did some initial, iterative annotation work on a subsample of the test data, which revealed which cases were not covered by the guidelines. These were the cases where the sentences were annotated with the Unsure label. From these uncovered cases, I revised the guidelines to cover them and then carried out another round of annotation work. This process of annotation and revision was repeated until all cases were covered. Finally, to estimate the quality of the annotations, I did a final round of annotation, and then calculated the intra-rater agreement (see section 3.13) between this final annotation iteration and the iteration before that. This gave a Cohen's kappa of $\kappa = .835$, which is a "Perfect" agreement according to the Landis and Koch scale of agreement, and thereby indicated that the annotation guidelines were of sufficiently good quality in terms of reliability.

Annotating Sentences

Using the developed annotation tool and following the annotation guidelines, I annotated 5,450 sentences over 5 weeks, carried out in parallel with the development of the classifiers. The number of annotated sentences was ultimately constrained by the amount of time available for annotation work.

Once the annotation period was over, I re-annotated a small subsample of 50 randomly selected sentences from the annotated sentences. These re-annotations were used for estimating the intra-rater agreement of the annotated sentences with Cohen's kappa (see section 3.13).

Cleaning Up and Splitting the Data

Once the annotation work was done, I cleaned up the data by removing the sentences annotated with None and Unsure. Duplicate sentences were also removed. The data was then split 50:50% into a dev set and a test set.

The dev set was intended for developing and adjusting hyperparameters of the classifiers. For the rule-based classifier, it was also used for training. To make the dev set suitable for training, I split it again, into a dev-train set (80%) and dev-test set (20%). These two sets were also separately upsampled so that the class frequencies were equal. Finally, I also upsampled the test set. Table 5.1 lists all the data sets and Figure 4.2 illustrates the splitting of them.

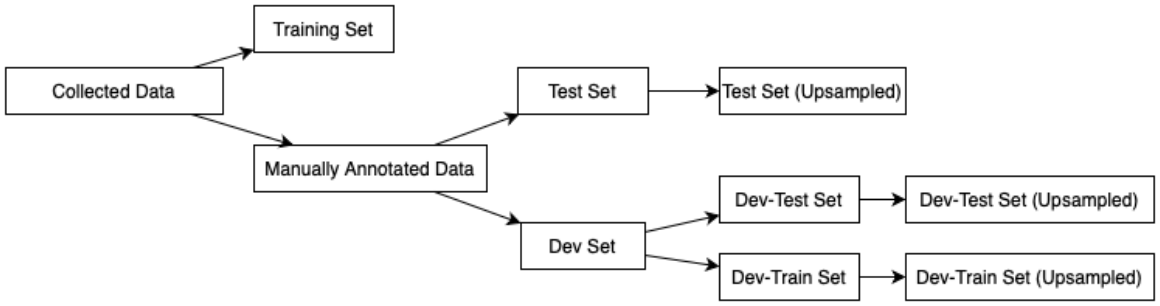


Figure 4.2: How the different data sets were split from the original set of collected data.

4.3 Developing the Rule-based Classifier

By using the development set, I developed and trained a rule-based classifier (see section 3.4). This classifier uses a basic system of rules that shallowly describe the syntactic structure of a sentence. Each rule is mapped to a speech act label, and the classifier's task is to find the rule that most closely matches the sentence to be classified. The rule that matches is used for classifying the sentence. A sentence can be classified as either an *Assertive*, *Question*, *Directive*, *Expressive*, or *None*. The classifier learned 98 rules.

Encoding the Relevant Syntactic Information

In more detail, the classifier receives a sentence to be classified. It begins by identifying the root word, that is, the word that is not dependent on any other word (see section 2.4 for syntactic dependencies). It then picks out the dependencies of the root word, that is, the words that are dependent on the root word. These words, including the root word, are then converted into what I refer to as *synt-blocks* (or syntactic blocks). These blocks each encode different kinds of grammatical information, for example, whether a word is a finite verb, or an interrogative pronoun, or a subject. Essentially, the synt-blocks encode the specific pieces of information that are relevant for determining the speech act of a sentence (see Section 2.2). A full list of all synt-blocks is given in Table 4.1 and 4.2. (Note that a synt-block does not just represent a single word, but also all the dependencies of that word, for example, it can represent a whole clause or phrase.)

Some words do not have a corresponding synt-block, and these are thereby not converted. For the converted words, however, their synt-blocks are placed into a sequence in

the order that their corresponding words appear in the sentence. For example, the sentence “I went home.” is converted into the synt-block sequence {SUBJECT, FINITE_VERB, PERIOD}. This sequence thus encodes shallow syntactic information about the sentence. This process is illustrated visually with an example in Appendix C.

It is in principle possible to expand the list of possible synt-blocks (table 4.1 and 4.2), but doing so would run the risk of the classifier overfitting to the data. Keeping the number of synt-blocks low, keeps the model’s complexity low.

Table 4.1: The synt-blocks in the classifier for words dependent on the root word.

Synt-block	Description
SUBJECT	Where the subject is placed hints at the clause type of the sentence.
SUBJECT_2ND	Second person subject such as *du*, *ni,* and *Ni*. This is relevant to directives.
INT_PRON	Interrogative pronoun. This is relevant to questions.
INT_ADV	Interrogative adverb. This is relevant to questions.
QUESTION_MARK	Question marks are common in questions.
PERIOD	Periods are common in assertives.
EXCLAMATION_MARK	Exclamation marks are common in expressives and directives.

Table 4.2: The synt-blocks in the classifier for root words.

Synt-block	Description
FIN_VERB	Finite verb. Where the finite verb is placed hints at the clause type of the sentence.
FIN_VERB_IMP	Finite verb in imperative. This is relevant to directives.
PART_VERB	The root is a participle.
SUP_VERB	The root word is a verb in supine form. Sometimes the finite verb “har” is omitted, and then this becomes the root.
ADVERB	The root is an adverb.
NOUN	The root is a noun.
ADJECTIVE	The root is an adjective.
NUMBER	The root is a number.
PROPN	The root is a proper noun.
NONE	The root word is not supported. This also applies to the root-dependent words.

Search for Matching Rules

Once the sentence has been converted into a sequence of synt-blocks, the sequence is then matched to a specific rule by the classifier. A rule also consists of a sequence of synt-blocks, which is mapped to a speech act. As an example:

IF {SUBJECT, FINITE_VERB, PERIOD} **THEN** Assertive

The sentence's sequence and the rule's sequence will match if they are identical. However, they do not necessarily have to be identical to match. Rather, the two match if all the following conditions are met:

1. The sequence for the sentence is shorter or equal in length to the sequence for the rule.

2. Every synt-block for the sentence must be paired up with an identical synt-block in the rule.

3. The synt-blocks in the sentence must come in the same order as in the rule.

These matching conditions mean that a sentence can be matched to several rules. For example, if a sentence has the synt-block sequence {SUBJECT, FIN_VERB}, then it can be matched by both the {SUBJECT, FIN_VERB} and {FIN_VERB} rule. The former rule is more specific (it encodes more information), and the latter less. When the classifier searches for a match, it begins with the most specific rules and proceeds to the less specific. In this way, it finds the first rule that matches most closely, but if no such is found, it can fall back on simpler, less specific rules. As an example of this, some of the last rules it checks concern only the punctuation of the sentence, as in, whether it ends with a period, question mark, or exclamation mark.

When a matching rule is finally found, the sentence is classified with the speech act of that rule. If no rule is found, however, the sentence is classified as None.

Training and Learning the Rules

To determine the rules, their synt-blocks, and what speech acts they produce, I trained the classifier on the dev-train data set. For the classifier to learn a rule, it would receive a sentence, convert it into a sequence of synt-blocks, and search for a matching rule. While this process is similar to the classification process described above, there is one major difference: two sequences match if and only if they are identical. If it does not find a matching rule, it then adds the unmatched sequence of synt-blocks and the speech act of the sentence as a new rule for the classifier.

However, if it does find a match, it will register the occurrence of that speech act for that rule. In other words, during training, the classifier keeps track of the class frequencies with regard to each rule. For example, a rule may match with 10 assertion sentences and 2 questions. The classifier counts these occurrences and increments them as new matches occur. When the training is completed, the classifier selects the speech act that occurs the most for each rule, and in this example, it would be the assertion since it occurs more than the question. In general, this method is referred to as *Maximum Likelihood Estimation* (see Jurafsky & Martin, 2023 for other applications of it), and it allows the classifier to solve ambiguities by selecting the most probable class. This is somewhat similar to the approach of using confidence and support in class association rule mining (see section 3.4).

Expressives and Sentiment

Another ambiguity the classifier faced was to differentiate assertives from expressives. As noted in section 2.2, expressives are often indirectly expressed as assertions ("This movie is great" as opposed to "What a great movie"). To mitigate this ambiguity, it proved useful to consider the sentiment of a sentence, that is, whether it expresses a positive, negative or neutral attitude (see section 3.7). To see this, I calculated the correlation between assertives/ expressives and neutral/non-neutral sentiment in the dev data set using the Phi coefficient. This showed that expressives are more likely to contain non-neutral sentiment than assertives $\phi = .39$, $p < .001$.

This fact is used by the classifier in the following way. For a sentence, the classifier first classifies it using the system of rules as described above. Then, if it is classified as an assertive, it will look at the sentiment of the sentence. If the assertive sentence then has a non-neutral sentiment, it will instead be classified as an expressive. On the other hand, if it has a neutral sentiment, it will remain an assertive.

4.4 Automatically Annotating the Training Set

The remaining sentences that were not extracted for manual annotation were used for the training set. I removed all duplicate sentences. In addition, if a sentence also occurred in the test set, then it was also removed. I used the rule-based classifier to automatically annotate them with speech acts. Sentences that were classified as None were removed. See Table 5.1 for the size of this data set.

4.5 Training the Embedding-based Neural Classifier

With the automatically annotated training data, I trained a neural classifier to classify sentences via SBERT sentence embedding. The sentences can be classified with the following speech acts: *Assertive*, *Question*, *Directive*, or *Expressive.* Unlike the rule-based classifier, it does not have a *None* class.

The Neural Architecture

The classifier consists of a sentence embedding layer and a linear classification layer (see Figure 4.3). A sentence is first fed to the embedding layer where it is computed into a sentence embedding (see section 3.7). This embedding layer is a Swedish SBERT transformer model, pre-trained by KBLab (Rekathati, 2021). After computing the embedding, it is fed to the classification layer, which is responsible for assigning a speech act to the embedding. The classification layer is a single, fully connected, linear layer, meaning it does not have any hidden layers or activation functions (see section 3.3). For the input embedding, it computes a score for each speech act. It then classifies the embedding with the speech act that receives the highest score. A similar setup was suggested by Jurafsky and Martin (2023) for the related task of dialogue act classification, except using BERT (presumably with average pooling) instead of SBERT. I chose SBERT instead because it produces semantically meaningful sentence embeddings (see section 3.7). For instance, the word "what" means a different thing in a question compared to expressive: "What is that car?" compared to "What a great car!".

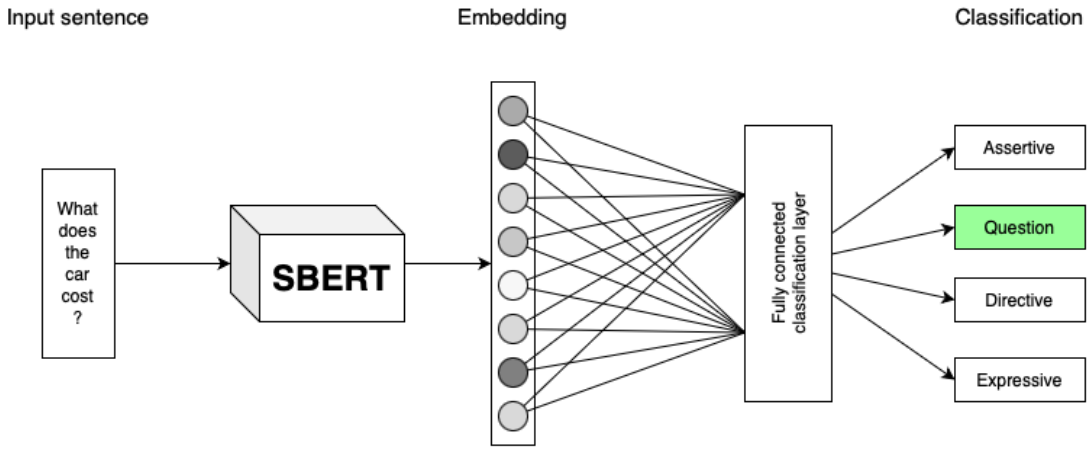


Figure 4.3: The embedding-based classifier consists of an SBERT embedding layer and a linear classification layer.

Training the Network

The training was done solely on the classification layer by back-propagating the loss gradients. The loss was computed with a categorical cross-entropy loss function. Since the training data was unbalanced, the loss calculation was done in conjunction with class weights to counteract the effects of the unbalanced classes (see section 3.5). I did this as an alternative to upsampling the training data, which would have led to a larger data set, and in turn longer training time.

The loss was then used for updating the weights. For this, I used the Adam optimizer as is often the recommended practice. I trained the network on the automatically annotated training data. This was done over 10 epochs, with a learning rate of .01, and a batch size of 16.

4.6 Evaluating and Comparing the Two Classifiers

Finally, I evaluated both the classifiers on the manually annotated test data. This was done by computing an accuracy score, recall, precision, F1-score, and an averaged F1-score (see section 3.2). These were compared to each other and a baseline. The baseline was computed by classifying all sentences as assertives (the specific class does not matter since the test set was balanced). A confusion matrix was also computed for each classifier.

Chapter 5
Results

Here I present the data sets that I created and the results from the evaluation of the classifiers.

5.1 The Data Sets

In total, I created 8 data sets, and these are listed in Table 5.1. Figure 5.1 illustrates the class frequencies for the test set and the training set. The test set together with the dev set have an intra-rater agreement of $\kappa = .87$. According to the Landis and Koch scale, this indicates a "Perfect" agreement.

Table 5.1: The different data sets that were created in this thesis.

Data set	Number of sentences	Annotated	Description
Training set	3,291,365	Automatic	For training the embedding-based classifier.
Test set	2,435	Manual	For final evaluation.
Test set (upsampled)	5,324		
Dev set	2,232	Manual	For developing the classifiers.
Dev-train set	1,787	Manual	For training the rule-based classifier.
Dev-train set (upsampled)	4,264		
Dev-test set	445	Manual	For developing the rule-based classifier.
Dev-test set (upsampled)	1,036		

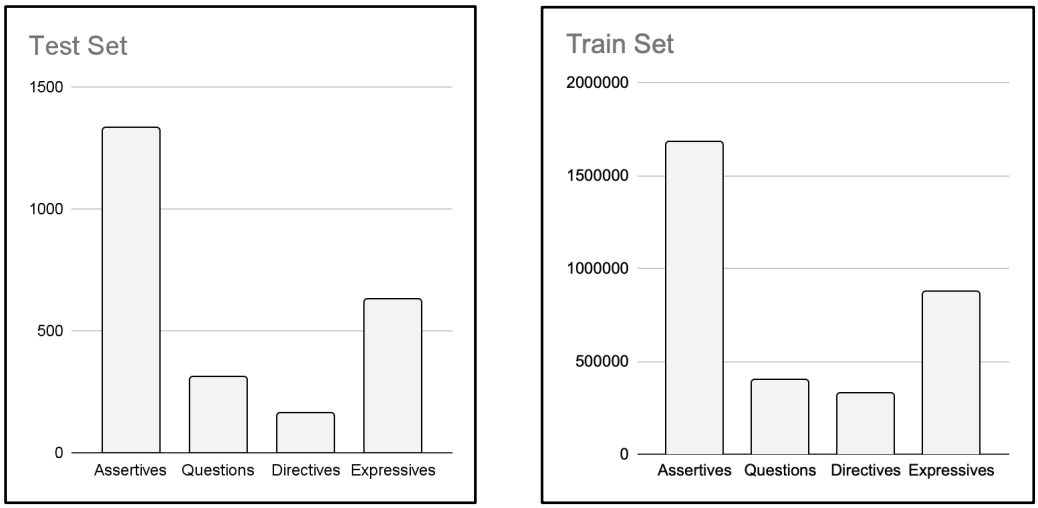


Figure 5.1: The number of words for each speech act label in the test set and the training set.

Data Format

For the annotated datasets, I used the CoNLL-U data format. For an example of a sentence, see table 5.2. In addition to the standard CoNLL-U annotations (Universal Dependencies, n.d.-a), I have added the following attributes as sentence comments to each sentence:

- *sent_id*: a unique identifying integer. This is unique across all the data sets.
- *text*: the full, unsegmented sentence.
- *date*: the date and time on which the sentence was posted on the internet forum.
- *url*: the URL of where the sentence was posted.
- *genre*: the text genre of the sentence. This is technically superfluous since all the sentences are of the same genre, namely *internet_forum*.
- *x_sent_id*: the ID of the sentence in the original corpus.
- *speech_act*: the annotated speech act of the sentence, whether automatically or manually annotated. The possible values are *assertion*[6], *question*, *directive*, and *expressive*.
- *sentiment_label*: the label denoting the sentiment of the sentence. This was automatically tagged by the sentiment tagger. The labels are either *positive*, *neutral*, or *negative*.
- *sentiment_score*: the estimated probability of the sentiment label. As with the sentiment label, this was also done by the sentiment tagger.

Table 5.2: An example of an annotated sentence in CoNLL-U.

```
# sent_id = 2200888
# text = Känns hoppfull med så många exempel.
# date = 2009-10-26 16:19:10
# url = http://www.familjeliv.se/forum/thread/48269320-bara-solsken-och-hopp/1#anchor-m3
# genre = internet_forum
# x_sent_id = 053044fa6
# speech_act = expressive
# sentiment_label = positive
# sentiment_score = 0.9705862402915955
1 Känns känna|kännas VERB VB Mood=Ind|Tense=Pres|VerbForm=Fin|Voice=Pass 0 root _ _
2 hoppfull hoppfull ADJ JJ Case=Nom|Definite=Ind|Degree=Pos|Gender=Com|Number=Sing 1 xcomp _ _
3 med med ADP PP _ 6 case _ _
4 så så ADVERB AB _ 5 advmod _ _
5 många _ ADJ JJ Case=Nom|Definite=Def,Ind|Degree=Pos|Gender=Com,Neut|Number=Plur 6 amod _ _
6 exempel exempel NOUN NN Case=Nom|Definite=Ind|Gender=Neut|Number=Plur 1 obl _ SpaceAfter=No
7 . _ PUNCT MAD _ 1 punct _ _
```

[6] "assertion" and not "assertive" is the correct label here. Early in the work I used the term "assertion", and it therefore became part of the data format. However, I later started using "assertive" instead.

5.2 Evaluation of the Classifiers

Overall Performance

As Table 5.3 illustrates, the embedding-based classifier achieves both the highest accuracy and F1-score. Both the rule-based and embedding-based classifiers achieve higher scores than the baseline. The embedding-based classifier has a .05 higher accuracy and a 0.4 higher F1-score than the rule-based.

Table 5.3: Accuracy and averaged F1-score for the classifiers and the baseline. The highest scores are underlined.

	Baseline	Rule	Embedding
Accuracy	.25	.69	<u>.74</u>
Averaged F1	.10	.70	<u>.74</u>

Class Specific Performance

Table 5.4 shows that the embedding-based classifier has higher scores in 8 out of 12 metrics. Notably is the difference in the F1-score for both directives and expressives, where the embedding-based classifier is .06 higher for both speech acts. However, this difference is not as pronounced for assertives, where the difference is .02. For questions we see the reverse, where the rule-based classifier instead is higher with .01. Figure 5.1 shows the confusion matrices for the classifiers.

Table 5.4: Classification metrics of the two classifiers. The highest score of each metric is underlined.

	Precision (Rule)	Precision (Embedding)	Recall (Rule)	Recall (Embedding)	F1-score (Rule)	F1-score (Embedding)
Assertive	.53	<u>.60</u>	<u>.74</u>	.70	.62	<u>.64</u>
Question	<u>.96</u>	.94	.92	<u>.93</u>	<u>.94</u>	.93
Directive	<u>.76</u>	.72	.60	<u>.75</u>	.67	<u>.73</u>
Expressive	.64	<u>.72</u>	.51	<u>.57</u>	.57	<u>.63</u>

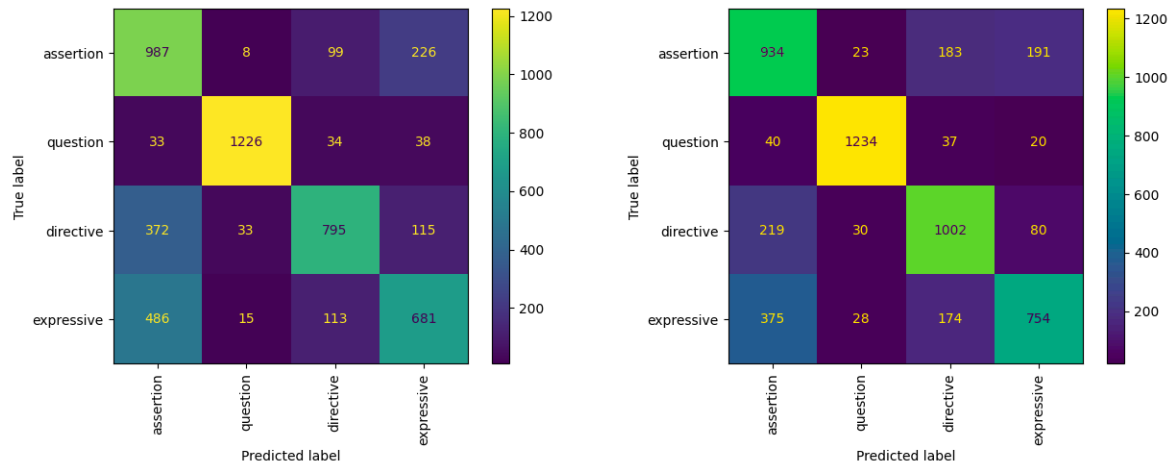


Figure 5.1: Confusion matrices for the rule-based classifier (left) and the embedding-based (right).

Example Sentences

Here I present some example sentences from the test set that were classified by the two classifiers. Table 5.5 shows some assertives that are correctly classified by both classifiers. It also shows that both classifiers can classify assertives that lack a subject.

Table 5.6 illustrates a variety of different expressives, some of which are correctly classified by the rule-based classifier and others by the embedding-based. Here, two of the expressives are direct speech acts, both of which are correctly classified by the rule-based classifier, but not by the embedding-based classifier.

As for directives, as illustrated in Table 5.7, both classifiers can identify direct and indirect directives.

Finally, as Table 5.8 illustrates, both classifiers can identify questions, however, the rule-based failed with the sentence where the question is given by a subordinate clause.

Table 5.5: Assertives. These are correctly classified by both classifiers. ✓ = correctly classified

Sentence	Rule	Embedding
Har också testat olika kablar och uttag.	✓	✓
Dock har han partiet emot sig.	✓	✓
Nu har vi alltid dörren på vid gavel där , så att det inte blir så fuktigt .	✓	✓
Så familjerätten har inte möjlighet att ta ett sånt beslut.	✓	✓

Table 5.6: Expressives. ✓ = correctly classified. - = misclassified. Direct speech acts are underlined.

Sentence	Rule	Embedding
Känns hoppfull med så många exempel.	✓	-
Dessutom är det en fantastiskt tid som aldrig någonsin kommer igen.	✓	-
Tackar tackar !	✓	✓
Så bra du är igång igen.	✓	-
Ska bli roligt att komma in här igen när jag kommer hem!	-	✓
Man vet aldrig vad han hittar på helt plötsligt !	✓	✓
Att han kom till slut!!!	✓	-

Table 5.7: Directives. ✓ = correctly classified. - = misclassified. Direct speech acts are underlined.

Sentence	Rule	Embedding
Sluta upp med att göra allting.	✓	✓
Läs först, svara sen =)	✓	✓
HJÄLP MEJ !!	✓	✓
Är nog det ni bör kolla.	-	✓
Audi A3/A4 TDI Quattro blir du nöjd med.	✓	✓
Du kan ju rensa tidigaremappen så får du bort det .	✓	✓
Vill du veta vilka företag som Estee Lauder samt Loreal äger så kan du gå in på dessa länkar:	✓	✓

Table 5.8: Questions. ✓ = correctly classified. - = misclassified.

Sentence	Rule	Embedding
I vilken stad?	✓	✓
Betalar jag dubbel hyra då, eller hur blir det?	✓	✓
Om man tar in växterna i huset kan man då räkna med ett gäng extra husdjur?	✓	✓
Vad ska jag göra ?	✓	✓
Antar du mena att man temporärt går över sitt genetiska max sålänge man kurar, men får man då efter att ha tränat nåra år ren efter massa kurande faktiskt behålla nån del, som är över sitt rena genetiska max?	-	✓
När tror ni att de nyantagna börjar söka lägenheter?	✓	✓

Chapter 6
Discussion

I will in this chapter answer the research questions of this thesis, as well as discuss some of its primary methodological limitations. I will also give a brief account of some ethical and environmental aspects concerning the technology involved.

6.1 Results

The main objective of this thesis was to develop an automatic classifier for speech acts. In doing so, I set out to answer three research questions concerning this objective. I will now answer these questions based on the results of this thesis.

Answering RQ1: Creating the Test Set

> **RQ1**: How can we create a test set of sentences annotated with speech acts for evaluating the performance of a classifier?

As for developing a test set, the results show that it can be done with the MATTER development cycle. Albeit, I have deviated on a few points from the actual cycle:

- **No explicit modeling**: According to the MATTER cycle, the linguistic phenomena should be turned into a formal model describing the components and features of the phenomenon and how they relate to each other (see section 3.12). This model is then used for creating the annotation guidelines. I have, however, not created such a model for speech acts. Instead, I have created the annotation guidelines directly from the theories, skipping the modeling step.

- **One-person annotator**: Typically, the annotation work is carried out by several annotators. I have, however, carried it out on my own. While the MATTER guidelines (Pustejovsky & Stubbs, 2013) do not explicitly oppose a single annotator, they do not recommend it either.

- **Single iteration**: MATTER is a cycle, meaning it should be carried out over several iterations, incrementally refining the annotated data. However, I have instead only carried out a single iteration.

Nevertheless, I still ended up producing a satisfactory data set. We see that it is satisfactory partially because of the “Perfect” intra-rater agreement, but mainly from the rule-based classifier's high performance when trained on a subset of this data (the dev-train set).

Answering RQ2: Semi-automatic Annotation of the Training Set

> **RQ2**: How can we use semi-automatic methods for creating a large training set of sentences annotated with speech acts?

For this, I developed and trained the rule-based classifier. The results show that it vastly outperforms the baseline on both accuracy and F1-score. The confusion matrix (see Figure 4.1) shows that it can differentiate between speech acts, albeit with some exceptions for directives and expressives. This is most likely because they are expressed as indirect speech acts. However, the results also show that it can correctly classify some indirect speech acts of these types. This suggests that syntax can be a good marker for speech acts—even indirect ones in some cases. It is true, however, that the rule-based classifier relies on sentiment, which is a form of semantics, and it is therefore not a purely syntactic classifier. But having said that, it relies on sentiment only for differentiating some expressives from assertives, far from all test cases. Hence, much can be done from syntax alone concerning speech acts.

This rule-based classifier was then used for automatically annotating a large training set of sentences. The classifier requires these sentences to be annotated with part-of-speech tags, lemmas, dependency relations, morphology, and sentiment—all of which can be done automatically. Hence, the process of creating a training set can in principle be fully automated —from data pre-processing to speech act annotation—with the only exception of actually training the rule-based classifier.

Answering RQ3: The Embedding-based Classifier

> **RQ3**: How can a pre-trained SBERT language model be used to classify speech acts of written Swedish sentences?

I used a Swedish SBERT model to compute the embeddings of the sentences, which were then classified with a single, fully connected, linear layer. The automatically annotated training set was used for training this classifier. The results show that this classifier achieved a higher accuracy and F1 than the rule-based classifier. The greatest improvements are for expressives and directives. This suggests that the embedding-based classifier has not only learned the semantic differences between speech acts but because of this, it is also able to better differentiate between them.

This also shows that the semi-supervised learning approach that I have taken is a viable method for creating an embedding-based classifier, that is, training a rule-based classifier on a small set of manually annotated sentences, and then using the classifier to automatically annotate a larger training set. Having said this, it is still important to consider that I have not compared this to a non-bootstrapped baseline. What performance would the embedding-based classifier achieve if it was instead trained only on the dev-train set? If the same, then bootstrapping would not make any contributions. Since I have not done such a comparison, it is not possible to conclude that the bootstrapping is improving the final performance. It can only be concluded that the knowledge in the rule-based classifier can be distilled into an embedding-based neural network classifier—going from syntax to semantics.

6.2 Implementation

This thesis involved several steps, from annotation to evaluation, and as such many potential sources of error stem purely from the methodological choices that I have taken. I will here present some of the primary methodological limitations of this thesis, as well as provide some alternative methods for better rectifying these.

Interpreting Speech Acts out of Context

Perhaps one of the main limitations of the methodology of this thesis is that the speech acts have only been analyzed in isolation from any dialogue. This is the case both for the manual annotation work and for the automatic classification. In natural conversation, we humans tend to interpret what is being said in the context of the dialogue, hence much of the utterance's meaning prevails in this context as opposed to in the utterance alone. This fact is put to use in the methodology of Conversation Analysis (see section 2.4). For this thesis, it would therefore have been more appropriate to adopt this approach instead. Both when manually annotating the sentences, and when training and testing the classifiers. However, this would probably increase the amount of time required for manual annotation, as well as increase the amount of data required to train the classifiers.

However, a possible consequence of this lack of context is that it harms the validity of the annotated speech acts. I may, for example, have annotated an indirect directive as a question or expressive, since I could not see its actual function in the conversation.

Spoken vs. Written Speech Acts

In section 2.1, I mentioned that speech acts occur on social media as well, but primarily in written form. Due to the restriction of non-verbal cues, people overcompensate in their messages by altering them to more strongly express affect, playfulness, and depth. Hence, written sentences in online communication differ from spoken face-to-face sentences. This may harm the generalizability of this thesis. Since the data in this thesis is only from online discussion forums, the classifiers may not apply as well to spoken communication. Therefore, the ideal approach would have been to create a data set from transcribed spoken communication.

The Speech Act Taxonomy

In this thesis, I used the speech act taxonomy by Teleman et al., (1999). However, Searle's (1979) taxonomy is far more established in the field of pragmatics. Why have I chosen Teleman over Searle? The sole reason is that they were easily available and well documented for Swedish in the *Swedish Academy Grammar* (Teleman et al., 1999). However, while the taxonomies are different, they do overlap to some extent. Much of what has been done in this thesis can be generalized to Searle's speech acts. However, care must be taken in doing so, because I have treated Searle's commissives and declaratives as assertives. For example, the declarative "You are fired" and the commissive "I will travel to Thailand this spring." would both be classified as assertives by the classifiers. This is also the case for the annotation guidelines.

An alternative approach for this thesis would therefore have been to acknowledge these speech acts, either by annotating sentences of these acts with the Other label (see section 4.2), or by adding them as labels to annotate. This would mean that the manually annotated data sets would be generalizable to Searle's taxonomy. However, the annotation work for me would become problematic due to the context-dependent nature of declaratives. Knowing whether an utterance is a declarative or not requires knowledge of the speaker and their authority regarding making such declarative acts (Searle, 1979). For example, an employee does not have the authority to fire his employer, hence his saying "You are fired" cannot be a declarative in this case. So, while such an approach would be feasible for commissives, it would not be for declaratives.

A further point is the granularity of the taxonomy. Why choose a taxonomy of only five speech acts, of which only four are used? The Switchboard corpus (Calhoun et al., 2010) consists of 42 different speech acts (or technically dialogue acts). So why not take a similar approach? The reason here is again the availability of Teleman's taxonomy—much of the work has already been done. To my knowledge, there is currently no highly granular speech act taxonomy with accompanying syntactic properties for Swedish sentences. Such a taxonomy would therefore have to be developed from scratch, which is beyond the scope of this thesis. More speech act categories would also require a longer time to develop the annotation guidelines as well as more annotation work. Hence, greater granularity would not be feasible for this thesis.

The Annotation Process

For annotating the sentences, I have followed the MATTER cycle (see section 3.12). However, as I have mentioned in section 6.1, I have deviated on at least three points from the MATTER standard. While still producing good results, it would perhaps have been ideal to not deviate at all, since adhering to an established standard increases the validity of the annotated data.

The first deviation was that I did not develop an explicit model of speech acts as a phenomenon, that is, how different linguistic properties of a sentence relate to each other and the different speech acts. Instead, the annotation guidelines were developed directly from the linguistic theories. In a sense, the guidelines can be viewed as the model, but this is not how it was intended in the MATTER cycle. Hence, for this thesis, I should instead have developed a proper, explicit model.

The second deviation was that I carried out the annotation work on my own, instead of training other annotators to do it. I had developed both a set of annotation guidelines and a web-browser-based annotation tool, so much of the necessary framework was already was already at hand. Therefore, for this thesis, it would have been ideal to train a group of annotators to annotate instead.

Finally, the third deviation was that I only carried out one iteration of the MATTER cycle. It would have been preferable to go through several iterations, further improving the annotation guidelines, the annotated data, and thereby the performance of both the classifiers —better data mean better classifiers.

Hence, in the best of all possible worlds, this thesis would have involved proper modeling of the speech acts, training several annotators, and carrying out several iterations of the MATTER cycle.

Developing the Rule-based Classifier

When developing the rule-based classifier, I have drawn inspiration from both Rule Induction systems and Class Association Rule Mining (see section 3.4). However, instead of using confidence and support for each rule, I have instead used maximum likelihood estimation (see section 4.3). This was primarily done because it was easier to implement and at the same time gave satisfactory results. Ideally, I should have used confidence and support, as well as minimum support and minimum confidence to prune rules during training. While it is not possible to estimate the effects of this choice, it would nonetheless be preferable to adhere to the established standards and technologies for rule-based classification.

Another point of concern is how I used sentiment, where I have used it as a post-processing step: first a sentence is classified using a rule and then if it is an assertive, I control for sentiment. An alternative approach would perhaps be to identify words with sentiment using a lexicon and then encode these words with a sentiment synt-block. This way, the sentiment would be part of the rules, as opposed to being a post-processing step. This would perhaps solve the issue with mixed sentiment (see section 3.7), where a sentence can express both positive and negative sentiment about something. It solves it by removing the distinction between positive and negative, distinguishing only between sentiment and no sentiment. Further, in such an approach, only shallow sentiment words (those occurring as dependents of the sentence's head word) are considered, ignoring sentiment found deeper in the syntactic tree. Whether this would improve classification performance, I do not know. However, a lexicon approach would relinquish the need for pre-tagging the sentiment of all the sentences. Considering that I did this using a Transformer model, a lexicon approach would be significantly cheaper concerning computation.

Sources of Error in the Classifier Performance

As I mentioned, better data mean better classifiers. While one source of error comes from the quality of the manual annotations, other sources stem from the different automatic processing stages of the data (see Figure 6.1).

1. In the original corpora, the linguistic annotations, such as segmentation, part-of-speech tags, and morphology, are automatically processed and annotated. This means the quality of the corpora is limited to the performance of the classifiers and segmentation processors that were used.

2. The original part-of-speech tags were converted to Universal Dependencies tags (see section 4.1). The conversion table used for this is not perfect but is only an approximation.

3. The quality of the dependency relations is limited to the performance of the automatic dependency parser in Stanza.

4. The quality of the sentiments is limited to the performance of the sentiment classifier. Furthermore, there is no "mixed" label for sentences with several sentiments (see Section 3.7). Instead, these sentences get classified as "neutral", thereby falsely giving the impression that they do not express any sentiment.

5. The rule-based classifier was used for automatically annotating the training data. It is of course limited by its features (the synt-blocks) as well as the dev-train set data it was trained on (see Sections 4.2 and 4.3).

6. The quality of the sentence embeddings is limited by the performance of the SBERT model.

7. Finally, the classification layer is limited by its architecture and its training. For example, what would happen if I had used a deeper network with hidden activations, instead of just a single linear layer? As for the training, factors that could affect the performance would be the choice of learning rate, the number of epochs, the weight optimizer, the batch size, and the loss function.

Further, these errors are not independent of each other, but rather the errors accumulate in further errors in each downstream step. For instance, the errors in the part-of-speech conversion affect the dependency parsing errors, which in turn, together with the errors of the sentiment classifier, affect the errors of the rule-based classifier. It is difficult to estimate to what degree these errors have on the final performance, but it is nonetheless worth acknowledging that they exist.

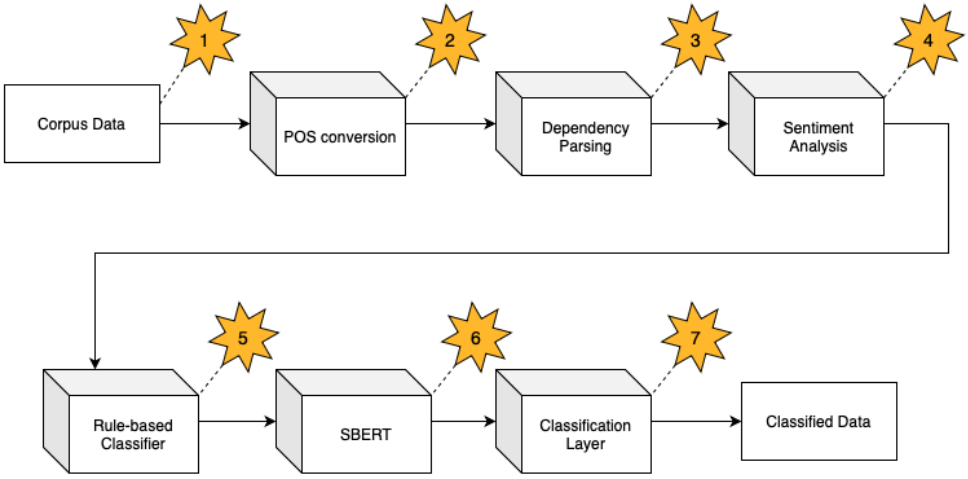


Figure 6.1: The sources of error in each processing step.

One potential source of error that could have been accounted for is the presence of invalid and broken sentences. These types of sentences occur in the original corpus most likely because of faulty sentence segmentation (see Appendix C for examples). For the manually annotated data, these were removed as part of the annotation process using the Other label (see section 4.2). However, there was no such cleaning process for the automatically annotated data (the training set). This means that there is a risk that the training set contains a lot of these broken sentences. If so, it would negatively impact the training of the embedding-based classifier and its performance. Hence, it would have been better if I had implemented some automatic cleaning process, perhaps as part of the rule-based classifier itself, or as a separate preprocessing step.

The Bootstrapping

In section 6.1 I mentioned that it is not possible to establish the effect of the bootstrapping—did bootstrapping make any contributions compared to instead training the embedding-based classifier on a sample of the manually annotated data? It would therefore have been preferable to train an embedding-based classifier on the dev-train set and use it as a baseline. Additionally, it would have been good to measure how the accuracy changes as the classifier is trained on more data. Suendermann et al., (2009) did this by plotting a graph of the accuracy as they received more and more data for their bootstrapped classifier. This graph shows how the accuracy increases as it is trained on more data. This is something I could have done as well since it would tell us if more data improves performance or not.

To summarize, the ideal setup for this thesis would involve:

- Using a Conversion Analysis approach to annotating speech acts, as well as using data from spoken conversations.
- Developing a highly granular speech act taxonomy which is at the same time compatible with Searle's (1979) taxonomy.
- Not deviating from the MATTER cycle, and instead explicitly modeling speech acts, training a group of annotators, and carrying out several iterations of the MATTER cycle.
- Using confidence and support values for each rule rather than maximum likelihood estimation.
- Encoding sentiment words as a particular synt-block using a lexicon approach.
- Using automatic data cleaning methods for the training set.
- Training a baseline embedding-based classifier using the dev-train set.
- Plotting the embedding-based classifiers' accuracy as a function of the amount of data it is trained on.

6.3 The Thesis in a Wider Context

This thesis has focused on automatically classifying Swedish, general speech acts in written sentences from online discussion forums. As I have noted, there are other automatic classifiers of speech acts (see section 3.9). These vary in the number of speech acts they classify, as well as their target applications. Some of them classify dialogue acts rather than speech acts. However, these are all for English. This thesis has therefore been an initial step towards introducing this type of classifier for speech acts in Swedish.

What applies to the speech act classifiers above also applies to the currently existing speech act corpora (see section 3.10). While I have developed a test set for evaluating

classifier performance, this test set is only an initial version. This is partly because I only carried a single iteration of the MATTER cycle, but also because of the underlying data. While the chosen speech act taxonomy may generalize to other taxonomies, it is not certain the specific type of data will. The data originate from online discussion forums. As mentioned in section 2.1, in written computer-mediated communication, people tend to overcompensate in their messages for the lack of non-verbal cues that are normally present in spoken face-to-face communication. Therefore, the underlying data may differ from that of spoken conversations. Establishing to what degree the data may differ would have to be the subject of future investigation.

The MATTER cycle (see section 3.12 and Pustejovsky & Stubbs, 2013) is still a relatively new methodology. Many data sets are annotated without this type of general convention. The annotation methodology is usually developed on the spot for the specific task at hand. Albeit, it may include conventional tools, such as inter-rater agreement, annotation guidelines, annotation software, and tag sets. However, it is usually up to the researcher to put these together into a complete methodology. The MATTER cycle, in contrast, is an already assembled methodology. The results in this thesis give some validity to MATTER as a methodology.

Finally, the classifiers developed in this thesis do not exist for their own sake. Rather, they can be used as tools for analyzing written language. In section 1.2 I mentioned corpus linguistics, where linguistic hypotheses can be tested on annotated corpora or to find new patterns in language. Li et al., (2017) found such patterns for speech acts using their manually annotated DailyDialog corpus. However, the dialogs they analyzed were written by human authors, so it brings into question the validity of these patterns and whether they apply to natural dialogs as well. The point is nonetheless that speech acts can be analyzed quantitively using corpora. Using classifiers, such as the ones developed in this thesis, these types of speech act corpora can be created automatically, similar to how corpora can be created automatically for parts-of-speech and syntactic structure. These speech act classifiers can therefore be a valuable tool in the computational linguist's toolbox.

6.4 Future Work

There are two paths one can take for future work. The first is to further develop the data set and the classifiers. The second is to apply the classifiers for corpus linguistic analysis.

To further develop the data set and the classifiers, one approach is to redo the work of this thesis but with the methodological improvements mentioned in section 6.2. Furthermore, additional iterations of the MATTER cycle can be done to improve the quality of the data. Also, a greater variety of data can be used, not limiting the data to only sentences from online discussion forums. In addition, as for the classifiers, they both rely on the Stanza pipeline for preprocessing but are not themselves integrated into the pipeline. The classifiers could be integrated by making them into custom processor variants (Pipeline and Processors, n.d.). This would make it easier to use the classifiers as tools.

The second path is to use the classifiers for linguistic analysis. To find conversational patterns in Swedish dialogues, similar to what Li et al., (2017) did with the DailyDialog data set. However, in doing so, one must keep in mind the methodological limitations which these classifiers rest upon (section 6.2), as well as that they are trained only on data from online discussion forums.

6.5 Ethical and Sustainability Considerations

While neural networks and large language models are in a sense abstract, mathematical models, they are implemented on physical hardware, and in turn, rely on large amounts of computations that require vast amounts of energy. The necessary production of this energy has both financial and environmental costs. Strubell et al., (2019) have shown that training a BERT model has roughly the same amount of carbon emissions as a trans-American flight. It is therefore important to consider the potential environmental impacts when using neural networks. This applies also to the embedding-based classifier that I have developed since it relies on an SBERT model. Hence, when classifying speech acts, it may not always be necessary to achieve a 73% accuracy; it is perhaps sufficient with the 69% accuracy of the rule-based classifier. Using the rule-based classifier requires far fewer computations, both for training and for inference. It does however require sentiment labels, and in this thesis, they were provided by a BERT model. But it is nonetheless possible to use other, less computationally heavy methods for sentiment analysis, for example, Naive Bayes classification (Jurafsky & Martin, 2023). While this does not provide state-of-the-art performance, it would nevertheless be a more environmentally friendly alternative.

Chapter 7
Conclusions

This thesis aimed to develop an automatic classifier for speech acts. This involved creating a test set of manually annotated sentences that could be used for evaluating classifiers. The data for this was retrieved from corpora which originate from online discussion forums. The annotation process followed the MATTER cycle (with some minor deviations) and the results indicate that this was a feasible approach.

Furthermore, a subset of these annotated sentences was used for developing and training a rule-based classifier. This classifier identifies the speech act of a sentence by analyzing its syntactical and grammatical features that are based on linguistic theories regarding speech acts. The evaluation of this classifier showed that it performed well above the baseline. Hence, for many sentences, the speech acts can be identified from syntax and sentiment alone.

Finally, the rule-based classifier was used for automatically annotating a large training set. This training set was then used for training a neural network to classify speech acts using sentence embeddings. In contrast to the rule-based classifier, this neural classifier identifies the speech act of a sentence by analyzing its semantic meaning—going from syntax to semantics. While this classifier had a higher performance than the rule-based classifier, it was not possible to conclude if this was because of the increase in data or because of its differing architecture.

Bibliography

Amini, M.-R., Feofanov, V., Pauletto, L., Hadjadj, L., Devijver, E., & Maximov, Y. (2023). *Self-Training: A Survey*. https://doi.org/10.48550/arXiv.2202.12040

Arguello, J., & Shaffer, K. (2015). Predicting Speech Acts in MOOC Forum Posts. *Proceedings of the International AAAI Conference on Web and Social Media, 9*(1), Article 1. https://doi.org/10.1609/icwsm.v9i1.14604

Buchholz, S., & Marsi, E. (2006, June 2006). *CoNLL-X shared task on Multilingual Dependency Parsing.* Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X), New York City. https://aclanthology.org/W06-2920.pdf

Carr, C. T., Schrock, D. B., & Dauterman, P. (2012). Speech Acts Within Facebook Status Messages. *Journal of Language and Social Psychology, 31*(2), 176–196. https://doi.org/10.1177/0261927X12438535

Chaka, C. (2020). Online Polylogues and the Speech Acts of Online Discussion Forums. *Journal of Educators Online, 17*(2). https://eric.ed.gov/?id=EJ1268925

Cohen, W. W., Carvalho, V. R., & Mitchell, T. M. (2004). Learning to Classify Email into "Speech Acts." In D. Lin & D. Wu (Eds.), *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (pp. 309–316). Association for Computational Linguistics. https://aclanthology.org/W04-3240

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297–334. https://doi.org/10.1007/BF02310555

De Marneffe, M.-C., Manning, C. D., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, 1–54. https://doi.org/10.1162/coli_a_00402

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics, 7*(1), 1–26. https://doi.org/10.1214/aos/1176344552

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*(5), 378–382. https://doi.org/10.1037/h0031619

Hägglöf, H. (2023). *The KBLab Blog: A robust, multi-label sentiment classifier for Swedish*. https://kb-labb.github.io/posts/2023-06-16-a-robust-multi-label-sentiment-classifier-for-swedish/

Joksimović, S., Jovanović, J., Kovanović, V., Gašević, D., Milikić, N., Zouaq, A., & van Staalduinen, J. P. (2020). Comprehensive Analysis of Discussion Forum Participation:

From Speech Acts to Discussion Dynamics and Course Outcomes. *IEEE Transactions on Learning Technologies, 13*(1), 38–51. https://doi.org/10.1109/TLT.2019.2916808

Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (3rd ed.).

Landauer, T. K. (2014). LSA as a Theory of Meaning. In *Handbook of Latent Semantic Analysis*. Routledge. https://doi.org/10.4324/9780203936399.ch1

Li, X.-L., & Liu, B. (2014). Rule-Based Classification. *In Data Classification*. Chapman and Hall/CRC.

Lindholm, A., Wahlström, N., Lindsten, F., & Schön, T. B. (2022). *Machine Learning: A First Course for Engineers and Scientists*. Cambridge University Press.

Liu, B., & Zhang, L. (2012). A Survey of Opinion Mining and Sentiment Analysis. In C. C. Aggarwal & C. Zhai (Eds.), *Mining Text Data* (pp. 415–463). Springer US. https://doi.org/10.1007/978-1-4614-3223-4_13

Krippendorff, K. (1970). Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement, 30*(1), 61–70. https://doi.org/10.1177/001316447003000105

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space* (arXiv:1301.3781). arXiv. http://arxiv.org/abs/1301.3781

Moldovan, C., Rus, V., & Graesser, A. C. (2011). Automated Speech Act Classification For Online Chat. *MAICS, 710*, 23–29.

Nivre, J., & Megyesi, B. (2007). Bootstrapping a Swedish Treebank Using Cross-Corpus Harmonization and Annotation Projection.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

*Pipeline and Processor*s. (n.d.). Stanza. Retrieved May 4, 2024, from https://stanfordnlp.github.io/stanza/pipeline.html

Pustejovsky, J., & Stubbs, A. (2013). *Natural Language Annotation for Machine Learning*. O'Reilly Media, Inc.

Russell, S., & Norvig, P. (2022). *Artificial Intelligence A Modern Approach* (4th ed., Global Edition). Pearson Education Limited.

Sacks, H., Schegloff, E. A., & Jefferson, G. (1974). A Simplest Systematics for the Organization of Turn-Taking for Conversation. *Language, 50*(4), 696–735. https://doi.org/10.2307/412243

Searle, J. R. (1969). *Speech Acts*: An Essay in the Philosophy of Language. Cambridge University Press.

Searle, J. R. (1979). *Expression and Meaning: Studies in the Theory of Speech Acts*. Cambridge University Press.

Sentiment—Amazon Comprehend. (n.d.). Retrieved April 24, 2024, from https://docs.aws.amazon.com/comprehend/latest/dg/how-sentiment.html#

Strubell, E., Ganesh, A., & McCallum, A. (2019). *Energy and Policy Considerations for Deep Learning in NLP* (arXiv:1906.02243). arXiv. https://doi.org/10.48550/arXiv.1906.02243

Suendermann, D., Evanini, K., Liscombe, J., Hunter, P., Dayanidhi, K., & Pieraccini, R. (2009). *From Rule-Based to Statistical Grammars: Continuous Improvement of Large-Scale Spoken Dialog Systems*. 4713–4716. https://doi.org/10.1109/ICASSP.2009.4960683

Teleman, U., Hellberg, S., & Andersson, E. (1999). *Svenska Akademiens grammatik* (1 ed., Vol. 4). Svenska Akademien.

Trevino, L. K., Lengel, R. H., & Daft, R. L. (1987). Media Symbolism, Media Richness, and Media Choice in Organizations: A Symbolic Interactionist Perspective. *Communication Research, 14*(5), 553–574. https://doi.org/10.1177/009365087014005006

Universal Dependencies. (n.d.-a). *CoNLL-U Format*. Retrieved 28-02-2024 from https://universaldependencies.org/format.html#conll-u-format

Universal Dependencies. (u.d.-b). *Tagset sv::suc conversion to universal POS tags and features*. Retrieved 28-02-2024 from https://universaldependencies.org/tagset-conversion/sv-suc-uposf.html

Universal Dependencies. (n.d.-c). Retrieved April 26, 2024, from https://universaldependencies.org/u/dep/all.html

Universal features. (n.d.). Retrieved April 27, 2024, from https://universaldependencies.org/u/feat/index.html

Universal POS tags. (n.d.). Retrieved April 27, 2024, from https://universaldependencies.org/u/pos/index.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. https://doi.org/10.48550/arXiv.1706.03762

Walther, J. B., & Burgoon, J. K. (1992). Relational Communication in Computer-Mediated Interaction. *Human Communication Research, 19*(1), 50–88. https://doi.org/10.1111/j.1468-2958.1992.tb00295.x

Yule, G. (1996). *Pragmatics*. Oxford University Press.

Appendix A

Språkbanken Text Corpora

The following table lists the corpora that were used.

	Corpus name	Date last updated
1	Flashback: Dator & IT	2024-02-20
2	Flashback: Droger	2024-02-22
3	Flashback: Ekonomi	2023-01-16
4	Flashback: Fordon & trafik	2023-01-26
5	Flashback: Hem, bostad & familj	2023-02-07
6	Flashback: Kultur & media	2023-01-18
7	Flashback: Livsstil	2023-01-19
8	Flashback: Mat, dryck & tobak	2024-02-19
9	Flashback: Om Flashback	2024-02-20
10	Flashback: Politik	2023-02-10
11	Flashback: Resor	2024-02-19
12	Flashback: Samhälle	2023-02-19
13	Flashback: Sex	2023-01-26
14	Flashback: Sport & träning	2023-02-14
15	Flashback: Vetenskap & humaniora	2024-02-26
16	Familjeliv: Adoption	2023-01-12
17	Familjeliv: Allmänna rubriker - Ekonomi & juridik	2023-01-12
18	Familjeliv: Allmänna rubriker - Familjeliv.se	2023-01-12
19	Familjeliv: Allmänna rubriker - Fritid & hobby	2023-01-13
20	Familjeliv: Allmänna rubriker - Hus & hem	2023-01-14
21	Familjeliv: Allmänna rubriker - Husdjur	2023-01-13
22	Familjeliv: Allmänna rubriker - Kropp och själ	2023-01-14
23	Familjeliv: Allmänna rubriker - Nöje	2023-01-15
24	Familjeliv: Allmänna rubriker - Samhälle	2023-01-26
25	Familjeliv: Allmänna rubriker - Sandlådan	2023-01-27
26	Familjeliv: Änglarum	2023-01-27
27	Familjeliv: Förälder	2023-02-03
28	Familjeliv: Fråga experten	2023-01-19
29	Familjeliv: Gravid	2023-02-04
30	Familjeliv: Känsliga rummet	2023-02-12
31	Familjeliv: Medlemstrådar - Allmänna	2023-02-01
32	Familjeliv: Medlemstrådar - Föräldrar	2023-02-15
33	Familjeliv: Medlemstrådar - Planerar barn	2023-01-22
34	Familjeliv: Medlemstrådar - Väntar barn	2023-01-24
35	Familjeliv: Pappagrupp	2023-01-11
36	Familjeliv: Planerar barn	2023-02-15
37	Familjeliv: Sex & samlevnad	2023-02-16
38	Familjeliv: Svårt att få barn	2023-02-17

Appendix B
The Annotation Guidelines

Introduction

Whenever we say something, we carry out a social action—we *do* something with the words we speak. These social actions are called *speech acts.* The purpose of these guidelines was to aid me in annotating sentences with their speech acts. However, that was in the past. Now their purpose is to document and report how I carried out this annotation work, in order for potential future annotators to perhaps replicate this.

Annotating Speech Act Labels

There are six speech act labels: *Assertive*, *Question*, *Directive*, *Expressive*, *Unsure*, and *Other* (see below for further details on each). These are annotated with a web-browser-based tool[7] and are done in sessions of 50 sentences at a time.

Label: Assertive

An Assertive is a sentence where the speaker holds the propositional content to be true to some degree. The expression of the content can be absolute (“Bilen är grå”), uncertain (“Bilen kanske är grå”), unknowing (“Jag vet inte om bilen är grå”), or anywhere in between.

Assertives often end with a period (“.”), but other punctuations can also be used.

Some examples:

- “Bilen var grå.”
- “Det var ju han som sa att han ville åka.”
- “Han sa att det var en fin bil.”
- “Instämmer om avataren.”
- “Men det ska nog lösa sig”
- “Det är nog så.”
- “Jag vet inte om han såg den.”

Label: Question

A Question is a sentence where the speaker requests information about whether the propositional content is true, or under what conditions it is true. There are different types of questions:

- *Yes/no question*: the speaker asks whether the propositional content is true (“Har du sett den nya filmen?”).
- *Searching question:* the speaker asks under what conditions the propositional content is true (“När går tåget?”).
- *Question mark*: the sentence is only a question mark (“?”) or several (“???”).

[7] Available on GitHub: https://github.com/Daniel-B-Tufvesson/speech-act-classifier

- *Rhetorical question:* the speaker knows the answer and is not actually requesting information from the listener.
- *Echo question:* the speaker repeats what was said before (or parts of it) as a way to request the listener to elaborate.

Questions often end with a question mark ("?"), but other punctuations can also be used.

Some examples:
- "Vill du se en film?"
- "Varför gick han hem?"
- "Hur gick det?"
- "Då är frågan bara vad man ska göra."
- "Va?"
- "Kalle?"
- "Jag undrar vilken fisk man ska köpa?"
- "Varför??"
- "Vem fan gör så?"

Label: Directive

A Directive is a sentence where the speaker tries to, through the speech act, get the listener to carry out the action that the sentence describes. This always concerns a future action of the listener (compare "Du ska göra detta" and "Du borde ha gjort detta"). Directives are often indirect, since it can sometimes be considered rude to give a direct request (compare "Jag undrar om man kan få lite salt." and "Skicka hit saltet!").

There are different types of directives:
- *Command*: the listener is not allowed to refuse.
- *Appeal:* the action is assumed to be in the listener's interest.
- *Offer:* the listener is allowed to refuse, but is not allowed to carry out the action without the directive.
- *Advice/recommendation*: the listener is allowed to refuse or act without the directive.

Directives often end with an exclamation mark ("!"), but other punctuations can also be used.

Some examples:
- "Hämta bilen imorgon!"
- "Kolla in denna sida."
- "Jag kan rekommendera en större mobil."
- "Om du gillar action-filmer så kan du se den nya Mission Impossible-filmen"
- "Har hört att det kan vara ganska riskfyllt med sådana investeringar, så du kanske bör läsa på om olika strategier innan du börjar."
- "Du är inbjuden till mitt bröllop."
- "Gå in i affären och fråga bara?"
- "Vill du skicka saltet?"
- "Men kan du inte ge dig?"
- "Här har du en intressant länk, som du kanske kan ha nytta av!"

Label: Expressive

An expressive is a sentence where the speaker expresses some feeling or emotional attitude about the propositional content. There are different types expressives:

- *Expressions of emotion:* the speaker expresses an emotional attitude towards the propositional content which the speaker also holds to be true. Some examples:
    - “Vad bra!”
    - “Vilken fin bil!”
    - “Det är en bra film.”
    - “Att du inte bara kan komma i tid!”
    - “Förlåt”
    - “Boken är svårbegriplig”
    - “Det är livsfarligt att låta politiker bestämma hur skattepengar ska spenderas!”
    - “Fy vad hemskt!”
    - “Det är helt ok.”
    - “Inget fel med det!”
- *Wishes:* the speaker holds the proposition not to be true, but wants it to be true. Some examples:
    - “Jag vill ha en bil!”
    - “Snus ska vara billigt!!”
    - “Vad fint det hade varit med en sådan!”
    - “Känner bara för att stanna hemma istället.”
- *Fears:* the speaker holds the proposition not to be true, and does not want it to be true. Some examples:
    - “Skulle ALDRIG välja att föda hemma.”
    - “Detta ska självklart inte ske.”
- *Greetings:* the speaker greets the listener. For example: “Hej!”, “Välkommen!”, “Adjö!”.
- *Congratulations:* the speaker expresses praise for an achievement or good wishes. For example: “Grattis!”, “Lycka till!”.
- *Thanks:* the speaker expresses gratitude.
- *Laughter:* the speaker expresses lively amusement or sometimes contempt or derision by laughing.
- *Emotional exclamations:* the speaker expresses an emotion through a short cry or remark. For example: "Va!", "NEJ!", "Usch", "Fy", "Aj", "Jaa!", “Fan”, “Herregud!”.

Expressives sometimes end with an exclamation mark (“!”), but other punctuations are also used.

Label: Unsure

The Unsure label should be used for ambiguous sentences. A sentence is ambiguous if it is not clear what speech act it has, or if it has several speech acts and it does not belong to any of them more than the others.

Some examples:

- “Apropå ingenting...”
- “Ärligt nu.”
- “O så ÄTER man kött o kykling o sådant !”
- “puff”
- “anime o manga kontakter och man kolla på de sj så”
- “Inte där kommer det hem fulla elektriker, så varför tar du upp så konstiga exempel som är dessutom kassa.” (Is this a question, expressive or assertive?)

Label: Other

The Other label is a garbage label and should be used on, for example, broken sentences, merged sentences, foreign sentences, emojis, text that are not sentences, computer-generated text, table data, etc. It is essentially used for texts that are not valid sentences. However, valid is here not meant as grammatically correct. Even ungrammatical sentences should be labeled with a proper speech act. The Other label is hence intended only for texts that are clearly not Swedish sentences, or where it is clear that the sentence segmentation is wrong (see below about broken and merged sentences).

Some examples:

- “FÖDDA..... Fyrabarnsmor, då 42 år (nära 43)... fick en son 051116 mobygirl, då 40 år ... fick en son 060122 Sannastina, då 41 år .... fick en son 061104 SusanneN, 42 år ..... fick en son 071018 Tittija, 42 år .... fick en dotter 080102 Spigge, då 40 år ....fick en son 080209 xanni, 41 år ....fick en dotter 080215 Svarten 42 år och 7 månader ..... fick en tjej 080710”
- “Midka skrev 2009-05-29 20:32:40 följande:Dorian Ertymexx skrev 2009-05-29 19:04:59 följande: Är det fysiskt möjligt för dig?”
- “BF 080808 ulle4, 41 år....”
- “21 september - Sagoskog”
- “Open up the control panel and search for plugins.”
- “(5 år)”
- “:D”
- “:-)”
- “😂”
- “!”
- “, eller vad vet jag”
- “(Självgående, svett-jocke osv) En gammal självgående”

Problematic and Special Cases

There are gray areas that can be problematic when annotating sentences. Here are some common problematic cases and how to handle these.

Complex Sentences with Multiple Speech Acts

Complex sentences, such as sentences with several main clauses or with subordinate clauses, should be labeled with the speaker’s intended speech act. As an example, the sentence “Jag såg den nya Spider-man trailern och undrar om du skulle vilja se den med mig?” is a complex

sentence consisting of both an Assertive and a Question. However, the Assertive only exists to give context to the Question, so the whole sentence should therefore be labeled as a Question. This involves some human intuition and interpretation. In uncertain cases, label it instead as Unsure.

Merged Sentences

Sometimes the automatic sentence segmenter fails to divide sentences from each other, and they end up as one. And sometimes people use commas (“,”) instead of periods (“.”) to separate sentences, which the sentence segmenter then fails to account for. Sometimes do not write any punctuation marks at the end of the sentence, which again confuses the automatic segmenter. Cases such as these should be labeled with Other.

Some examples:

- “Innan detta va jag väldigt hypokondrisk.Tänk om jag får en hjärtinfarkt.”
- “Jag är ju ingen expert på hur mycket de olika företagen säljer, så om du menar vem som är världsledande i.o.m vems som säljer flest produkter så kan jag inte svara på det.Jag kan ju tala om vilka märken som jag och mina kollegor använder mest och vad som dominerar i press och annan media.”
- “Jag gick i skogen, Bladen hade börjat komma, Sen såg man många som tränade,”

Broken Sentences

Sometimes the automatic sentence segmenter has divided sentences where they should not have been divided. These should be classified as Other.

Some examples:

- “Apropå ingenting...”
- “(Självgående, svett-jocke osv) En gammal självgående”
- “Ärligt nu.”
- “, eller vad vet jag.”
- “) men inte på länge.”

Foreign Language and Mixed Language

As mentioned earlier, sentences written in a foreign language should be labeled with Other. However, sometimes sentences are mixed with both Swedish and foreign words. If more than 50% of the words in a sentence are foreign, then it should be labeled with Other. Otherwise, it should be labeled with its appropriate speech act.

Some examples:

- “Den nya filmen var awesome!” (Should be labeled with Expressive.)
- “This resulted in a sample representative of the population.” (Should be labeled with Other.)

Links, Dates, Numbers, and Names

Some sentences consist of only a web link or a name. These should be labeled as an Assertive. This also applies to non-random numbers (“10” and “2”, but not “48749187419”) and dates (“2017-08-22”). There is of course the exception if there is a question mark at the end (“Henrik?”, “15?”), in which case the sentence should be labeled as a Question.

Some examples:

- “10”
- “https://www.google.com/”
- “2017-08-22”
- “Lotta”
- “New York”

The Difference Between Unsure and Other

It is worth noting that sentences labeled as Unsure and Other are not to be included in the final data set. As a consequence, there is not much reason to differentiate these two. The Unsure label exists only for diagnostic purposes since it shows what sentences are not covered by the guidelines. This was only relevant for the development of the guidelines, but not for the final data set.

Appendix C:

Dependency Relations to Synt-block Sequence

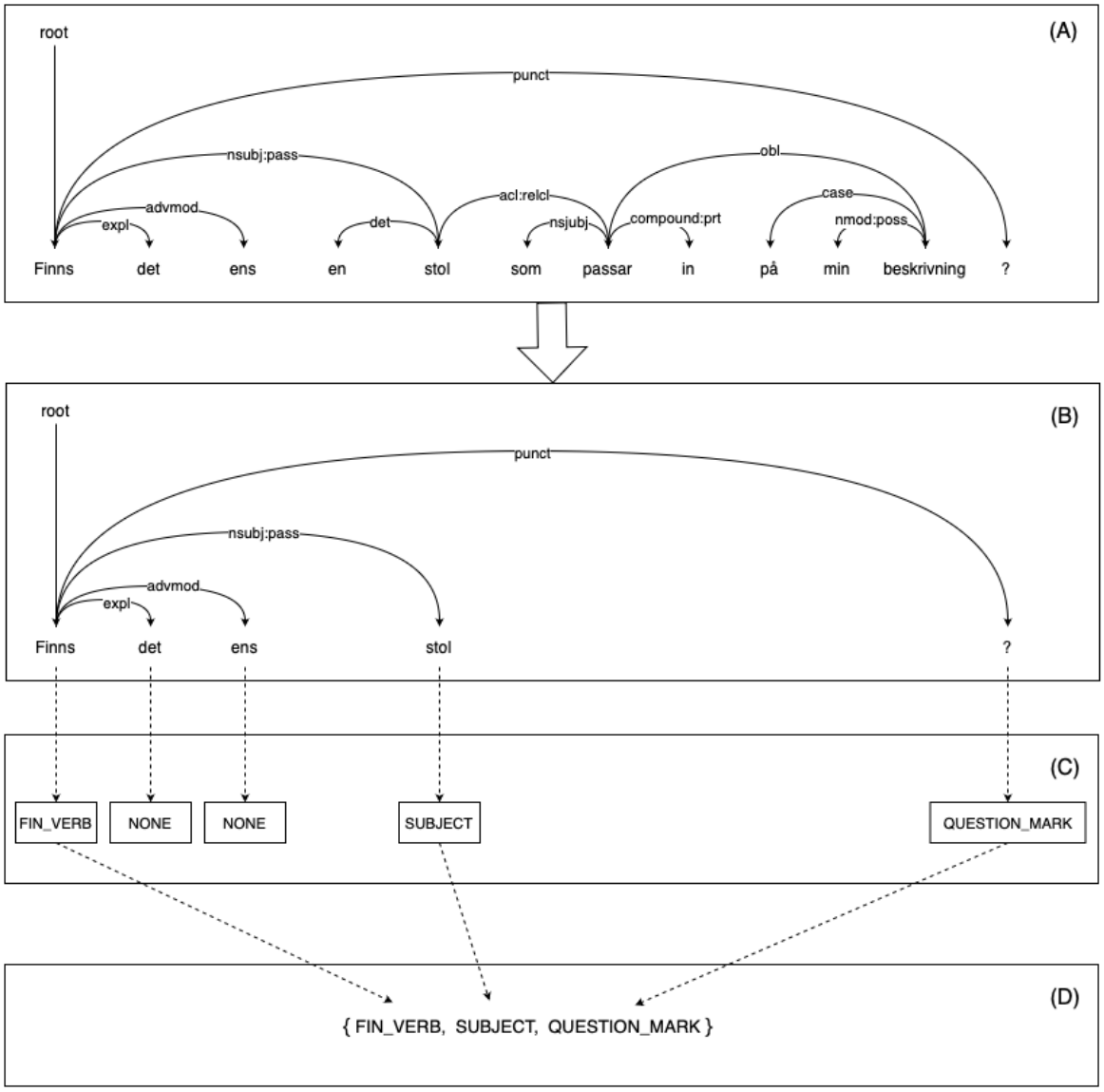
This is an illustration of how the rule-based classifier (see section 4.3) converts a sentence with dependency relations into a sequence of synt-blocks.

(A) The classifier receives a sentence and its dependency relations.

(B) The root word and its immediate dependent words are extracted.

(C) These words are converted into synt-blocks.

(D) The synt-blocks are placed in a sequence. The NONE blocks are omitted.



Appendix D:
The Learned Rules

Here, I list the rules that the rule-based classifier learned from the dev-train data (see Section 4.3).

	Synt-blocks	Predicted speech act
1	INT_ADV, FIN_VERB, SUBJECT, SUBJECT, QUESTION_MARK	question
2	INT_PRON, FIN_VERB, SUBJECT, QUESTION_MARK	question
3	SUBJECT, SUBJECT, FIN_VERB, PERIOD	expressive
4	INT_ADV, FIN_VERB, SUBJECT, QUESTION_MARK	question
5	INT_ADV, FIN_VERB, SUBJECT_2ND, QUESTION_MARK	question
6	SUBJECT, SUBJECT, FIN_VERB, EXCLAMATION_MARK	question
7	FIN_VERB, SUBJECT, SUBJECT, QUESTION_MARK	question
8	SUBJECT, FIN_VERB, SUBJECT, PERIOD	assertion
9	SUBJECT, SUBJECT_2ND, FIN_VERB, EXCLAMATION_MARK	directive
10	FIN_VERB, SUBJECT, SUBJECT, PERIOD	directive
11	SUBJECT, FIN_VERB, SUBJECT_2ND, PERIOD	assertion
12	INT_PRON, FIN_VERB, SUBJECT_2ND, QUESTION_MARK	question
13	SUBJECT, SUBJECT, PART_VERB, PERIOD	assertion
14	INT_ADV, FIN_VERB, SUBJECT_2ND, PERIOD	question
15	INT_PRON, ADJECTIVE, SUBJECT, EXCLAMATION_MARK	expressive
16	INT_PRON, ADJECTIVE, SUBJECT, PERIOD	expressive
17	SUBJECT, FIN_VERB, SUBJECT, EXCLAMATION_MARK	expressive
18	INT_ADV, SUBJECT, FIN_VERB, PERIOD	assertion
19	FIN_VERB, SUBJECT, SUBJECT, EXCLAMATION_MARK	assertion
20	FIN_VERB, SUBJECT, INT_PRON, PERIOD	assertion
21	FIN_VERB, SUBJECT, PERIOD	assertion
22	SUBJECT_2ND, FIN_VERB, PERIOD	directive
23	FIN_VERB, SUBJECT_2ND, PERIOD	directive
24	SUBJECT_2ND, FIN_VERB, QUESTION_MARK	question
25	SUBJECT, FIN_VERB, PERIOD	assertion
26	FIN_VERB, SUBJECT, EXCLAMATION_MARK	directive
27	FIN_VERB, SUBJECT_2ND, QUESTION_MARK	question
28	SUBJECT, FIN_VERB, QUESTION_MARK	question
29	FIN_VERB, SUBJECT, QUESTION_MARK	question
30	FIN_VERB, SUBJECT_2ND, EXCLAMATION_MARK	directive
31	SUBJECT, FIN_VERB, EXCLAMATION_MARK	expressive
32	SUBJECT, SUBJECT, PERIOD	assertion
33	FIN_VERB_IMP, SUBJECT, PERIOD	directive
34	SUBJECT, ADJECTIVE, PERIOD	expressive
35	SUBJECT, SUP_VERB, QUESTION_MARK	question
36	INT_PRON, NOUN, PERIOD	expressive

	Synt-blocks	Predicted speech act
37	ADJECTIVE, SUBJECT, PERIOD	expressive
38	INT_PRON, ADJECTIVE, QUESTION_MARK	question
39	SUBJECT, FIN_VERB_IMP, EXCLAMATION_MARK	expressive
40	SUBJECT_2ND, FIN_VERB, EXCLAMATION_MARK	expressive
41	INT_ADV, ADVERB, QUESTION_MARK	question
42	INT_PRON, FIN_VERB, QUESTION_MARK	question
43	SUP_VERB, SUBJECT, PERIOD	expressive
44	ADVERB, SUBJECT, PERIOD	expressive
45	INT_ADV, FIN_VERB, QUESTION_MARK	question
46	SUBJECT, NOUN, PERIOD	expressive
47	SUBJECT, SUP_VERB, PERIOD	assertion
48	ADJECTIVE, SUBJECT, EXCLAMATION_MARK	expressive
49	SUBJECT, ADJECTIVE, SUBJECT	expressive
50	INT_PRON, ADJECTIVE, EXCLAMATION_MARK	expressive
51	PART_VERB, SUBJECT, PERIOD	expressive
52	SUBJECT, PART_VERB, PERIOD	assertion
53	ADVERB, SUBJECT, SUBJECT	expressive
54	SUBJECT, PART_VERB, EXCLAMATION_MARK	assertion
55	SUBJECT, FIN_VERB, SUBJECT	assertion
56	INT_ADV, SUBJECT, FIN_VERB	assertion
57	SUBJECT, FIN_VERB	assertion
58	ADJECTIVE, QUESTION_MARK	question
59	FIN_VERB, PERIOD	assertion
60	FIN_VERB, SUBJECT_2ND	directive
61	NOUN, PERIOD	assertion
62	FIN_VERB_IMP, QUESTION_MARK	question
63	NOUN, EXCLAMATION_MARK	expressive
64	FIN_VERB, EXCLAMATION_MARK	expressive
65	FIN_VERB, SUBJECT	expressive
66	FIN_VERB_IMP, PERIOD	directive
67	SUP_VERB, QUESTION_MARK	question
68	ADJECTIVE, EXCLAMATION_MARK	expressive
69	NOUN, QUESTION_MARK	question
70	FIN_VERB, QUESTION_MARK	question
71	FIN_VERB_IMP, EXCLAMATION_MARK	directive
72	PROPN, PERIOD	directive
73	SUBJECT, PERIOD	assertion
74	ADVERB, QUESTION_MARK	question
75	ADVERB, EXCLAMATION_MARK	directive
76	PART_VERB, PERIOD	assertion
77	PROPN, QUESTION_MARK	question
78	SUBJECT, NOUN	expressive
79	SUP_VERB, PERIOD	assertion
80	ADJECTIVE, PERIOD	expressive
81	NUMBER, QUESTION_MARK	question
82	NUMBER, PERIOD	assertion
83	PART_VERB, QUESTION_MARK	question

	Synt-blocks	Predicted speech act
84	ADVERB, SUBJECT	expressive
85	PART_VERB, EXCLAMATION_MARK	expressive
86	ADVERB, PERIOD	assertion
87	PROPN, EXCLAMATION_MARK	expressive
88	NOUN	assertion
89	PERIOD	directive
90	EXCLAMATION_MARK	expressive
91	QUESTION_MARK	question
92	ADVERB	directive
93	FIN_VERB	expressive
94	FIN_VERB_IMP	directive
95	ADJECTIVE	expressive
96	PROPN	expressive
97	PART_VERB	expressive
98	NUMBER	assertion