Radboud University Nijmegen

Faculty of Social Sciences

Automating the Detection of Dispositional and Behavioural Phrasing

The Linguistic Category Model Applied to Dutch Job Advertisements

Thesis MSc Artificial Intelligence: Intelligent Technology

Author:
Jetske Adams

Internal supervisor:
Martha Larson

Second reader:
Iris Hendrickx

External supervisor:
Kyrill Poelmans

January 24, 2022

Abstract

Measures are taken in companies worldwide to improve their DEIB strategies and attract more diverse groups of employees. Creating an inviting atmosphere starts at the job advertisement. One contributing factor is how on one hand the job's activities and on the other hand the job's person requirements are worded: behavioural wording (concrete descriptions of the job including actions and details) has been linked to being generally understood as being less prone to subjective judgement than dispositional wording (abstract descriptions of desired characteristics of the applicant). The unintentional use of negative meta-stereotypes, which are stereotypes that members of social out-groups expect can be held against them, is expected to be hidden in dispositional wording. This idea relates to the Construal Level Theory (CLT), which poses that the mental representation people have of a subject becomes more abstract with an increased psychological distance towards it. In this thesis, the Linguistic Category Model (LCM) is applied to detect dispositionally and behaviourally phrased predicates through various subcategories, namely 'Act', 'Process', 'Attitude + action', 'Attitude', 'Innate quality', and 'Learned quality'. An annotation guide was written and improved by running five annotation pilots. The final guide was used to generate an annotated dataset of predicates labeled according to the subcategories. Two approaches were investigated for automating the detection of dispositional and behavioural predicates in text. The first approach was three-step sequence tagging, where predicate boundaries were predicted with a rule-based system and the predicates were classified with a Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, Gradient boosting, LSTM on Word2Vec embeddings, and fine-tuned model of BERT and RoBERTa. The predicates were first classified by relevance and then the relevant predicates were classified by their LCM category. BERT and RoBERTa gave the highest accuracy. The second approach was one-step sequence tagging, where labels were predicted in-text on token level with a fine-tuned model of BERT. After applying both approaches on two example texts, it appeared that the three-step sequence tagging model is better at predicting correct predicate boundaries, while the one-step sequence tagging model is suggested to give more accurate word-level label predictions. The word-level class prediction accuracy was respectively .69 and .77. Automating the detection and labeling of predicates as defined by the LCM opens the opportunity for sociologists and psychologists to conduct a wide range of studies. In the domain of job advertisements, it would help to locate possibly excluding phrasing and to quantify the job advertisement's level of abstraction automatically. The presented approaches may inspire to develop tools that increase awareness of writers of job advertisements about their language use.

Contents

1 Introduction

According to CBS's annual report on Integration, on the 1st of January 2020, almost a quarter of the Dutch population had a migrant background, half of whom had a non-western migrant background. Around 63% of them had a job, while for people with a Dutch background this was 70% (Swagerman, 2020). Although employment rates of the former group have been on the rise over the last few years, there still is an employment gap between ethnic majority and minority groups in the Dutch society. Not to mention discrimination in the workplace itself, which might enter the floor at any later stage. It can therefore be more challenging for those with a foreign background to get hired for a job and to keep a job. With an increasing ethnic diversity in the Dutch population, recognizing this issue is more crucial than ever before.

To improve accessibility of the job market for ethnic minorities, many factors come into play. This varies from writing the job advertisement, to the decisions made about which applicants are invited for an interview, to the process of actual hiring of new employees. This thesis proposes a way to analyse the very first step of the employment process: the formulation of the job advertisement. Bias can be found in many linguistic forms and structures. The scope of the problem is too large to capture and solve in one thesis and would need further social experimenting to be validated. Therefore, this is an attempt to detect and investigate phrasing of a more general type: abstract and concrete phrasing in job advertisement, with attention to the way in which the job-specific tasks and activities are described (behavioural wording) as well as how person requirements for the applicant are described (dispositional wording).

Research has found that certain abstract, dispositional terms can be associated with (negative) meta-stereotypes of minority groups and lowers their application rate as a result. By using these terms, people of ethnic minority backgrounds are thus placed at a disadvantage. Therefore, providing a measure for the 'abstractness' of the job advertisement and cues on where to find this type of phrasing, is the first step to help to detect and mitigate possibly exclusive language in this direction. This thesis will not provide the solution to this problem, but rather will propose general ways to automate the detection of dispositional and behavioural wording. This can be used in a range of studies to compare texts in any domain and increase awareness about the effect different types of phrasing have on a reader. It can also be built on when studying job advertisements, to provide guidance to recognizing in- and exclusive language.

In this thesis, the following steps are taken to detect dispositional and behavioural wording. The Linguistic Category Model (LCM) is used as a base to define subcategories of the two types of wording, on a scale from concrete to abstract predicates. The predicates that belong to any of the LCM subcategories are to be detected. To be able to train models for this task, an annotated dataset is needed. Several annotation pilot runs are required to evaluate and optimize the annotation guide used to obtain a set of reliable annotations. Two approaches were suggested to automate the detection of dispositional and behavioural wording: the first is three-step sequence tagging, and the second is one-step sequence tagging. The three-step approach separates the task. It consists of a predicate segmentation step to find all predicates with a rule-based method. The predicates are classified by relevance (a predicate is relevant if it fits into one of the LCM subcategories). Finally, the relevant predicates are classified either as one of the subcategories or as dispositional or behavioural. The one-step approach combines the steps of the three-step approach by classifying all tokens of a text as an LCM subcategory or as not relevant. The two approaches' strengths and weaknesses are compared.

1.1 Textmetrics

This thesis was carried out at Textmetrics, a company that provides an augmented writing platform for companies both in the Netherlands and internationally to help them compose and improve job advertisement posts by offering corrections and suggestions in the text. Textmetrics provided the data that was used for this thesis and the help necessary for data annotation. A main issue they try to target at the moment is improving the inclusivity of job advertisement in any dimension that people can feel discriminated against, for example on the basis of age, gender, ethnicity, and religion. This thesis is written to contribute to this ongoing process by providing ideas that can help make writers of job advertisements more aware of the message they convey and how it may be interpreted, with the ultimate goal of making the job market more welcoming towards social and/or ethnic minorities.

1.2 Goal

To get a deeper understanding of the inclusivity of a job advertisement, one important aspect to investigate closely is the use of behavioural vs. dispositional phrasing. This distinction is more elaborately reflected in the 'abstractness' of the text by mapping it to the Linguistic Category Model's subcategories. In job advertisements, the relevant sections of the texts that provide the information needed are the description of what the job entails and the description of the asked person requirements of the applicant in particular. In this thesis, therefore, the main goal is to detect dispositional and behavioural predicates in job advertisements automatically through subcategories that are based on those defined by the Linguistic Category Model.

This research offers a foundation for sociologists that want to study the effect of behavioural and dispositional wording on marginalized groups on a large scale. For Textmetrics, it would be helpful to develop an automatic method to highlights those parts in their job advertisement texts that may contain (meta-)stereotypes or need to be written in more concrete way to enable companies to attract the optimal group of applicants.

1.3 Research Questions

Having defined the goal, it gives rise to the following research question:

How can behavioural and dispositional wording be automatically detected in job advertisements?

This research question is divided into the following sub-questions:

1. *How to find the relevant sections in the text that are on the behavioural to dispositional spectrum?*
2. *How to create a protocol to annotate these identified sections consistently?*
3. *How to automate the process of detecting dispositional and behavioural phrases?*

The research questions will be addressed as follows.

Chapter 2 explains the details of the theory behind inclusivity, computational methods used in detection of discrimination, and how the Linguistic Category Model and the Construal Level Theory play a role in this. Chapter 3 describes the process of developing the annotation guide and how the final annotated dataset was made. Chapter 4 describes the annotated dataset. These chapters are focused on defining the problem space and obtaining reliable and usable annotations. This contributes to answering the first and second sub-questions.

Chapter 5 explains how the two approaches that are investigated are implemented and presents a way to quantify the texts' level of abstraction through the LCM score. In chapter 6, the results of each of the classifiers as well as the results of the three-step and one-step sequence tagging approaches are discussed. In chapter 7, it is compared how the two approaches perform in practice on two example texts. These chapters will comment on the third sub-question, which is concerned with how to automate the detection task (and to what extent this is possible).

Finally, chapter 8 concludes the thesis by revisiting the research question and its sub-questions, and offering suggestions and ideas for improvements and future work. In the appendices, the annotation pilots are reviewed in detail. They also contain the final version of the annotation guide and the results of the two automated tagging approaches on two example texts.

2 Theoretical framework

Diversity and Inclusion measures are taken seriously these days and companies worldwide have been working on diversifying their workplace over the last years to attract a wider variety of applicants and be able to call themselves inclusive organizations. With that, research on both explicit and implicit discrimination in the workplace has flourished. The use of Artificial Intelligence in this research, however, is still rather scarce. This chapter will first provide an overview of the meaning and importance of bringing inclusivity measures into the workplace. Then, it will get into the recent advances of how AI has helped to uncover discriminating biases in general. Lastly, this background theory will be connected to the Construal Level Theory and the Linguistic Category Model to find a way that the concepts may be combined and applied in the recruitment process.

2.1 Inclusivity on the job market

An inclusive work environment is inherently anti-racist. Especially in a multicultural country like the Netherlands, this means that people of all different cultural backgrounds mix, meet and work together, as a diverse group. DiB 2018 presents the image below to visualize inclusion. Inclusion is about more than creating a diverse work environment, it is about making everyone *feel* equally respected and valued and providing equal treatment (Diversiteit in Bedrijf, 2018). "Diversity is being invited to the party; inclusion is being asked to dance" (Witkamp et al., 2018). That means that active encouragement of in-group members towards out-group members to participate equally in society is vital to bridge this gap. Moreover, 'D&I' has since been extended with 'Equity' and 'Belonging', together forming 'DEIB'. These values stress that inclusivity means that everyone should feel welcome exactly *as they are*.

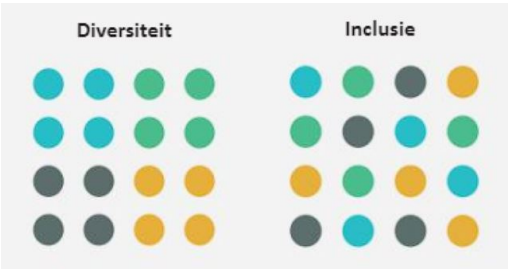


Figure 1: Difference between diversity and inclusivity

The very first step in providing equal chances to all on the job market is the formulation of the job description. That means, writing inclusive job advertisements according to the values expressed by DEIB. The text cannot contain any form of discrimination, and above that, it should be written in an inviting tone of voice for all groups in society. That includes factors such as how formal or informal the company is described, how much jargon is used, and how the reader is addressed (as "he"/"she"/"they" or "you").

Having some bias towards social groups in society is human, because everyone uses social generalization to some extent. Attributing certain traits and characteristics to social groups that one does not belong to (so-called 'out-groups') helps to provide structure in a way to make sense of the world around them, to find common ground when talking about 'us' and 'them'. Categorizing people into such groups - thereby often making

cultural generalizations - makes it possible to process the vast amount of (new) information we face every day. But this generalization carries a risk of enforcing unwanted stereotypes on individuals. The resulting prejudices made are often quite persistent and can lead to an effect called 'out-group homogeneity', which stands for the involuntary assumption that out-group members are more similar to each other than in-group members. Still, in-group members are seen as more similar to each other than to their out-groups. Because of this, people are naturally inclined to prefer members of their own social group, which is called the 'similarity-attraction' problem. This problem occurs also in organizations, according to the 'Attraction-Selection-Attricion' (ASA) theory. Another example of a phenomena that often occurs as an effect of this problem, is 'explaining away'. This happens when an out-group member's success is explained as them being an exceptional case of the social group they belong to, thus explaining their success as originating from an external factor or luck rather than believing that it could naturally happen due to the skill-set of the individual. As a consequence, the stereotype and corresponding prejudices are preserved.

The idea of similarity between social groups in the population was also referred to by Hagendoom and Hraba (1987) as 'intergroup social distance'. It is a form of ingroup-outgroup differentiation, as shown in figure 2, where this differentiation is said to be a form of racism. They also pose that out-groups can be positioned at different social distances from the in-group. That means that effects observed in one out-group may not be observed (to the same extent) in another out-group. The figure shows this by visually placing out-groups at different distances from the in-group. Out-groups can be socially closer to each other, if, for example, they share a dominant religion or political situation.

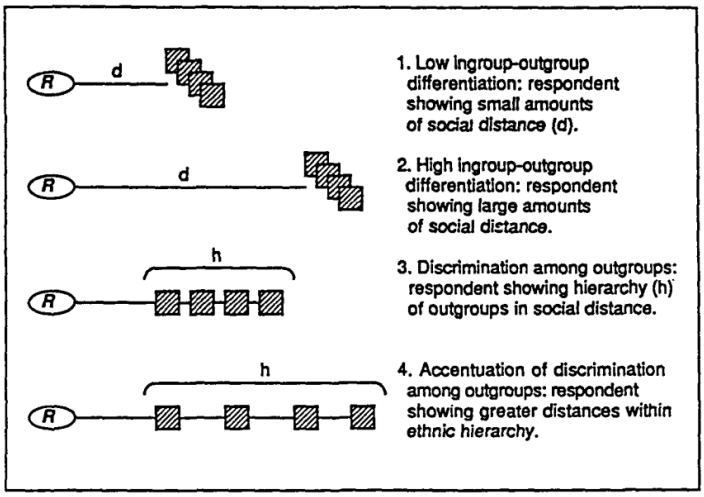


Figure 2: Types of ingroup-outgroup differentiation as measured between an in-group and any given ethnic out-groups (Hagendoom & Hraba, 1987). R represents the in-group.

An out-group may also have stereotypes for another out-group, shown as the out-groups that are placed farther apart on the social distance scale. In that case, they speak of an ethnic hierarchy. The discrimination that follows from this ethnic hierarchy is more subtle. They found that even in a low degree of discrimination, where it is not led by prejudices, it is still felt by minority groups in situations of more private or intimate contact.

Cultural stereotypes are reflected in language. Vorauer et al. (2000) introduced the term meta-stereotype, which is a term for what stereotype individuals tend to expect members of their out-group to attribute to them. This phenomenon makes talking about inclusion even more tricky. It is hard to distinguish 'neutral' terms from meta-stereotypes. Some examples are given by Wille and Derous (2017) for negative meta-stereotypes, which are meta-stereotypes with a negative association attached to them. These are present in words such as 'integrity', 'trustworthy', and 'reliable', indicating that people with a minority ethnic background might feel held off when encountering these terms because they expect that their potential employers could hold these stereotypes against them. It was also offered that a word like 'reliable' might be read more as a personal judgement than when phrased behaviourally like "you are expected to keep confidential information to yourself". That means that the way a trait is phrased matters. Although the words given as examples above are not negative by definition, they were found to lower the application rate of people with another ethnic background than the in-group when stated as a person requirement in the job advertisement. It could be more accessible and therefore more inclusive to use behavioural task-related wording rather than dispositional wording which targets the reader's personality.

To find out which stereotypes are attributed to which social groups, the Hofstede's theory might be of interest (Hofstede, 2021). This theory is used to give country scores on six dimensions of national culture: individualism, power distance, masculinity, uncertainty avoidance, long-term orientation, and indulgence. The country scores give away the strength of these dimensions in the countries' national cultures and can thus be compared (Hofstede Insights, 2021). Although the theory is generalizing and does not consider within-country variation, these cultural dimensions can give away overarching stereotypical traits of different social out-groups and how they interpret and are affected by ideas and language provided by other nationalities and/or cultures.

A big challenge to exclusive language is that biased phrases can be subtle and hard to notice. Whether they are acceptable or not depends on the context as well. However, the effects can be persistent and therefore it is especially important to find a balance between on one hand detecting a wide range of possibly exclusive phrases and on the other hand finding only the ones that actually have the effect of excluding people.

2.2 General computational methods of uncovering bias

The topic of bias detection in textual data is not a particularly new area of research. It has been ongoing for around 20 years. With the rise of algorithmic decision making, algorithmic bias has become a hot topic. How can a machine be expected to make a fair decision, if the data that it learns from is not fair or representative? Following the "garbage in, garbage out" principle, (implicit) bias in text will end up reflecting the decisions that are made and increase social polarization, instead of providing the objective judgement that is probably one of the main reasons to involve a machine in this process at all. This bias might also be found when investigating job advertisements. Discriminatory phrases and bias on the basis of cultural and ethnic background and religion could be found by looking for keywords, keyphrases, and other contextual cues. When detected, this bias should ideally be either removed or neutralized in the data, such that subsequent steps (for example filtering suitable job placements) are performed on a more unbiased basis. This can entail both a detection task and a suggestion-providing task. A more recent development, though, is trying to detect bias with the help of

machine learning. This only showed up during the last few years, as it became possible with the introduction of NLP and transformer models like ELMo and BERT using word embeddings to make sense of the meaning of texts. This section will provide a general overview on the progression of research in the field.

One method used for uncovering stereotypes is the use of word embeddings. It started with the analogy "man is to computer programmer as woman is to homemaker" (Bolukbasi et al., 2016), where it was discovered that through addition and subtraction of word embedding vectors, the relation between the word 'homemaker' and 'woman' was similar to that between 'man' and 'computer programmer'. A similar experiment was done on the Reddit data from users from the USA to uncover racial and religious analogies. They found that, for example 'black' was associated to 'homeless' as 'white' was to 'servicemen', whereas 'jew' was associated to 'greedy' as 'christian' was to 'intellectually' (Manzini et al., 2019).

Garg et al. (2018) have tried a similar experiment to find gender and ethnic stereotypes by using a Word2Vec model on the Google News Dataset that contains articles of the last 100 years as well as GloVe embeddings from the New York Times annotated corpus. They found that ethnicity (typical Hispanic/Asian/White last names) was associated with different occupations. Also, words related to violence correlated with words related to the Islam. Over all, the occupation biases in the embeddings tracked the actual occupation participation rates. This kind of bias may amplify systemic racism, whether it happens unconsciously or not. On the other hand, more specifically about Indonesian job advertisement texts, an attempt was made to identify direct types of discrimination using Direct Discrimination Detection (DDD) (Ningrum et al., 2020). They used a Discriminatory Keyword Dictionary (DKD) and Word Pattern Templates (WPTs) to show combinations of keywords and ordering of phrases. They found that direct discrimination on the basis of religion was present in 1.09% of the advertisements.

Another study developed the WEAT, the Word-Embedded Association Test, which is a statististical test applied to word embeddings, where the distance between a pair of vectors was found to be analogous to reaction time in the Implicit Association Test (Caliskan et al., 2016). They found that every human stereotype was replicated in the data (flowers were more pleasant than insects. Likewise, European-American names were found to be more pleasant than more typical African-American names). They also discuss the Sapir-Whorf hypothesis, which states that behaviour can be driven by the cultural history embedded in a term's historic use, and that such histories vary across languages, as languages vary across cultures. Thus, use of the same word can evoke different reactions and/or behaviours when read by people with different cultural backgrounds.

A more recent family of machine learning architectures are Transformers, amongst which the most famous ones are BERT, ELMo, and GPT. They are used for a wide variety of tasks. For example, ELMo was used to find occupation words that co-occur with gender revealing pronouns (Zhao et al., 2019). This bias was found in co-reference resolution. They showed that mitigating this type of bias can be done both by augmentation of data as well as by embedding neutralizing terms. The augmentation technique was most effective.

Another idea is to do Perturbation Sensitivity Analysis to test for unwanted biases, where linguistic phrases used for a group of people are substituted in the same linguistic context. This can be done for gender (gender-swapping) but also for different types

of disability groups (Hutchinson et al., 2020) or named entity mentions (Prabhakaran et al., 2019). By predicting contextual toxicity and sentiment for every swapped substitution, one variant could be recommended over another. BERT was used to show this.

This section has made clear how bias has been studied over the last years. The studies that make use of Natural Language Processing (NLP) methods have, for the most part, focused on specific word use. Occupations were related to pronouns, genders, or ethnicities. Ways to 'neutralize' biased data were proposed using data augmentation methods. Bias towards American-European or Western European ethnicities was found by most studies. What is missing, however, is a link relating discrimination in textual data to more implicit types of discrimination found in language. That is, in language that does not use outright exclusive words or analogies, but rather phrases a message (that is not intended to be exclusive) in a way that is perceived to be more welcoming when read by an in-group member than when read by an out-group member. The LCM and the CLT are introduced next to substantiate a way to recognize such language use.

2.3 The Construal Level Theory (CLT)

As seen in figure 2, one could state that everyone from a different ethnic group is at a further social distance from them than those people in the same ethnic group. This idea can be applied to any other social dimension, for example gender, sexuality, or educational background. What it comes down to is the idea that you are socially 'closer' to the people that you have more in common with. Social psychologist have found a theory for this which states that the abstraction level of the mental representation people internally create about each other, correlates with the psychological distance that they perceive between each other. The construal-level theory of psychological distance describes the relation between this psychological distance and the extent to which people's thinking is abstract or concrete (Snefjella & Kuperman, 2015).

The construal levels are found on a continuum from low-level to high-level construal. At the extremes of this continuum, the low-level construals are concrete, detailed, unstructured/incoherent, and goal irrelevant, while the high-level construals are abstract, emphasizing central features, are structured/coherent, and goal relevant. Information of high-level construals is also more stable over varying distances. Low- and high-level construals can be differentiated by asking the questions "how?" (the means) and "why?" (the goal) (Smith, 2014).

To illustrate this, look at the following sentences:

- He puts one foot in front of the other
- He walks to the station
- He is going home

In this example, the first sentence concretely describes the means, the action, of walking, to ultimately fulfil some goal that is not given. This is therefore a low-level construal. The last sentence describes the goal itself, getting home, but not the actions or the type of transportation that are used to get there. This is the abstract representation of the action, thus a high-level construal. The middle sentence has some of both, walking as action and the station as partial goal, so it is in the middle of the continuum. By asking "how?", the construal level of the phrase is lowered (move one sentence up in the example above). The other way around, asking "why?" can raise the construal level, but this is not always the case. Asking "why?" for the middle example sentence,

can result in the last sentence, which is more abstract. But it can also result in "He is going to catch a train", which is not necessarily a more or less abstract answer. Asking these questions can help to identify the construal level of a text.

Power is an influential interpersonal phenomenon. It is having control over someone else. It is also relative between individuals. Low-power people have less control but are instead more dependent on others, which leads to an increased asymmetric social psychological distance between high- and low-power people, where high-power people feel more distanced from low-power people than vice versa (Magee & Smith, 2013). As a result, high-power people are less likely to show incentive to affiliate with lower-power people, and are more likely to talk about low-power people abstractly. On the contrary, low-power people feel less distanced to the higher-power people. With a smaller perceived distance, the construal level that is used to describe the other group decreases.

When viewing those people who are affected by discrimination in job descriptions (often unemployed members of the out-groups) as a low-power group of people, that means that according to this theory they feel less psychological distance to the in-group than the in-group feels towards them. However, this feeling is not accommodated by the employer when abstract language is used in interpersonal communication, as the out-group still feels excluded more quickly than low-power in-group members.

Moreover, it was stated that being made to think more abstractly makes one feel more powerful (Wakslak et al., 2014) but being made to think more concretely does not make one feel *that* much less powerful. Thus, it is more about the mind-set that is created than the actual implied power difference. It has also been found that negative abstractions are conceived as criticism (just like meta-stereotypes). Describing other people in an abstract way implies a passing of judgement towards them and their character. Negative abstractions thus imply unwanted personal characteristics, whereas positive abstractions imply wanted characteristics. Thus, describing people in abstract fashion can be seen as being powerful because it implies having a judgemental view. It could be because of the flexibility that the abstract phrasing has. There is more room for different interpretations for abstract phrases than for concrete phrases so that means that you have some *power over* filling in the details that are not given.

Since having a high-power mind-set apparently activates abstract thinking, a reader may perceive the writer of a relatively abstract text as having a higher degree of power. So, there is a bigger psychological power distance when abstract language is used than when more concrete language is used. Applying this theory to the relationship between job-seeker and those in charge of hiring people at a company, the job-seekers may feel that a job is written more on their same power level instead of a higher power level if the advertisement is written using concrete language (using low-level construals). In that case, the text might be more accessible. On the other hand, according to the Hofstede's theory, different cultures have different preferences in Hofstede's dimension of interest. One dimension is called 'power distance'. The Netherlands scores low on this dimension, so hierarchies in the workplaces are relatively flat, and therefore a low power distance, as is implied using low-level construals, is thought to be preferred. However, foreign job seekers may feel more comfortable with a power distance that matches to their original country's power distance standards.

What the construal level theory comes down to is that the use of high-construals implies that the writer is powerful, and creates a bigger power distance from the reader to the writer as well as a feeling of passing judgement. If this abstract (high-level

construal) language is related to a negative stereotype, out-group members are likely more susceptible to this than in-group members. A relatively abstract job description could also give the suggestion of a stronger hierarchy within the company. Whether this is always desirable may be questioned, but if this 'abstractness' is possible to detect and/or quantify, the effect can more easily be mitigated when wished.

2.4 The Linguistic Category Model (LCM)

As found by Wille and Derous (2017), dispositional wording can have the effect of being too general, enforcing negative meta-stereotypes which can have the effect that out-group job-seekers do not feel inclined to apply for a job they are perfectly qualified for. This idea finds its roots in the Construal Level Theory, and can be applied using the Linguistic Category Model, proposed by Semin and Fiedler (1991). The LCM is a model to define interpersonal phrases on a scale from abstract to concrete. Even though job advertisements do not provide means of direct talking between the recruiter and job-seeker, they are a means of communicating a message to a reader, where the formulation of the message directly influences the interpretation of the reader. In that sense, job advertisements also contain a form of interpersonal communication.

Interpersonal terms consist of both verbs and adjectives, where verbs are used to describe actions and psychological states, and adjectives are used to describe traits or dispositions, which are personal properties of a person (Semin & Fiedler, 1991). More recent studies also include nouns and adverbs in the equation.

According to the Linguistic Category Model, interpersonal phrases can be divided into 5 categories: DAV, IAV, SV, SAV, and ADJ. These are defined as follows:

- DAV: Descriptive Action Verbs, refers to actions as observable activities with a clear beginning and end. These actions have a "physically invariant feature", that is, the action itself is set but the object and situation are context-bound. For example, 'walking' is done using your legs, so legs are the physically invariant feature that is shared between cases where 'walking' occurs. In general, DAVs do not have a positive or negative valence.
  Examples: [subj] *kisses* [obj], [subj] *kicked* [obj], [subj] *pushed* [obj].

- IAV: Interpretative Action Verbs, refers to verbs that denote actions that are not easily visualised. The actions belonging to this class have a defined beginning and end and positive or negative evaluative value. They "refer to a multitude of behaviors or actions that have the same meaning but do not share an invariant physical aspect" (Coenen et al., 2006).
  Examples: [subj] *helps* [obj], [subj] *cheats* [obj], [subj] *imitates* [obj]

- SV: State Verbs, refer to psychological (mental/cognitive or emotional/affective) enduring states, not evoked by a specific action or event. "They don't have a clearly defined beginning and end and cannot be objectively verified" (Coenen et al., 2006). State Verbs are said to be either stimulus-experiencer ("John amazes Mary") or experiencer-stimulus verbs ("John likes Mary"). In this example, amazement is caused by a specific action whereas liking is not, therefore only experiencer-stimulus is seen as SV.
  Examples: [subj] *likes* [obj], [subj] *hates* [obj], [subj] *respects* [obj].

- SAV: State Action Verbs, refers to another type of State Verb, the stimulus-experiencer type, which denotes the consequence of an action rather than the action itself like in IAV. "They refer to a general group of behaviors with a clearly

defined beginning and end, and have a positive or negative evaluative value." (Coenen et al., 2006).
Examples: [subj] *surprises* [obj], [subj] *amazes* [obj], [subj] *angers* [obj].

- ADJ: "Abstract inferences about the psychological features or characteristics of a person ... don't describe what a person does, feels, or thinks, but what a person is like" (Coenen et al., 2006). Some nouns and adverbs also have the quality that they can express what a person is like. Adjectives, can be differentiated by whether they have a morphological origin in a verb stem, and divided into the classes DAV-DAVADJ (talk-talkative), IAV-IAVADJ (help-helpful), SAV-SAVADJ (repulse-repulsive), SV-SVADJ (like-likeable), and genuine adjectives (extraverted, friendly, etc.).

The different verb types also range from phrasing concrete to abstract ideas, respectively in the order DAV, IAV, S(A)V, ADJ. That means that adjectives are most abstract type and descriptive action verbs are most concrete. This distinction was made under the assumption that abstract words are unrelated to events and say something about stable qualities of a person (e.g. "extraverted", "friendly"). This relates to the idea of dispositional wording, carrying information about enduring qualities of people, which overlap with known meta-stereotypes. On the other side, there are DAVs, which refer to specific behaviours of performing an action, without a contributing semantic positive or negative valence. They are observations rather than traits, which are more verifiable and disputable. For entirely unbiased writing, this means that the wished desired behaviour of a job candidate may therefore be best described by concrete language, rather than being hidden implicitly in a trait that could be interpreted differently than intended by an out-group member. While this theory is about interpersonal terms, it is used in wider domains. For example, in word of mouth to describe products (where the use of an adjective, such as in "the shirt is of bad quality", is seen as more abstract compared to the use of a verb, such as in "the shirt has faded". The second phrase can be derived by asking "why?" for the first phrase.) (Schellekens et al., 2010).

A related distinction can be made for extraversion and language abstraction, found by Beukeboom et al. (2013). They applied the Linguistic Category Model by analyzing face-to-face interactions between participants of their study. It was found that extraversion is linked more to abstract language while introverts use a lower level of abstraction and would rather stick to stating facts. They describe the most concrete linguistic category to be action verbs and the most abstract category to contain state verbs and adjectives.

The process of computing the abstractness/concreteness of a text has been automatized in various ways. Originally, human judges rated natural language use on abstractness, but the project by (Seih et al., 2017) computerized the assessment process based on a LCM verb dictionary and part-of-speech tagging to find adjectives and nouns. They found that writing in third-person contains higher levels of abstraction than writing in first-person. The most abstract category - adjectives - describes personality traits. It is most abstract because traits cannot be objectively examined (e.g. "he is charming" does not contain any information to validate the statement). Language abstraction becomes an indicator to assess to what degree people reveal their abstract or concrete thoughts to others in language use. This way, and supported by the construal level theory, psychological distance can be measured through language use, because people generally use more abstract language when talking about more 'distant' entities. Thus, people tend to generalize more when describing others than when describing themselves. The standard LCM coding procedure relies on a human-based coding strategy. This has

the downsides of being a cognitively demanding task for annotators, and as a result it is a time-consuming process and only a small set of annotated data can be obtained. They used Linguistic Inquiry and Word Count (LIWC) to store and count the dictionary found. With part-of-speech (POS) tagging, they identified verbs, adjectives and nouns. For each category of the LCM, they made a separate dictionary. To build the dictionaries, the data they used consisted of their own text corpus and also the General Inquirer LCM dictionary. Coders classified the verbs as DAV/IAV or SV and they ended up with a dictionary of 7489 verbs. Using the formula shown below, they found LCM scores for the texts in their data set.

$$LCMscore = \frac{DAV * 1 + IAV * 2 + SV * 3 + adjective * 4 + noun * 5}{DAV + IAV + SV + adjective + noun}$$

The LCM scores were significantly higher for text written in third-person than those written in first-person, which meant that participants writing or speaking in third-person used more abstract language.

Johnson-Grey et al. (2020) took another approach to measuring abstraction. They used hand-coded scores (BKW rating) instead of creating dictionaries for the LCM categories to quantify abstraction. This means that they rated the abstraction level of individual words on a scale from 1 (abstract) to 5 (concrete). E.g. "ethical" was rated 1.3 and "bald" was rated 4.69. Then they could find the abstraction score of sentences by averaging the word scores. They emphasise that the relationship between words is important to take into account for measuring abstraction besides the abstraction level of individual words.

In the research discussed above, the limitation of applying the LCM seems to be one that needs to link words to labels. In most studies, the same words are always linked to the same label or abstraction level. This would mean that the problem can be solved using a dictionary approach, where words or phrases are directly given labels. This approach has the advantage that it is straightforward to apply and understand. Words can be added and deleted at any time. However, it would also mean that the dictionary needs to be complete and contain all forms of the words to be identified. Because the scope of advertisements is so big, a big amount of data would be needed to capture all relevant information and to make it fit specific cases and jargon. It could work well on a general level, though, as was mentioned earlier in research by (Seih et al., 2017). Another limitation to take into account is that the LIWC (Linguistic Inquiry and Word Count) program that is freely available, which already provides a large part of the needed dictionary, is only available in English. Following the Sapir-Whorf theory, it is not guaranteed that translation of the LIWC dictionary into another language will correctly carry over the words' meanings in the context of the LCM, as each language has its own culture and history attached to its vocabulary.

3 Data annotation

To the best of our knowledge, automatic methods for detecting phrases of the LCM in texts have so far only been applied to English texts. Since any supervised classification model needs examples to learn from, an annotated dataset had to be generated out of sentences taken from Dutch job advertisements. In order to obtain reliable and consistent annotations, an annotation guide was written. This was done in a cycle of five annotation pilots, where annotators were given the same task and texts to annotate, and the Inter Annotator Agreement (IAA) was measured. After five pilots, this IAA score was high enough to be considered reliable and to continue generating the final annotated dataset. This chapter will first describe how the LCM labels were adapted to the domain of job advertisements. Then, it will discuss how the IAA was measured as well as the annotation tool that was used for generating the annotations. Then, the process of writing and improving the annotation guide is discussed, and lastly, it is explained how the final dataset was created.

3.1 Adapting the LCM to job advertisement text

The Linguistic Category Model is a model describing ways of communication in the interpersonal domain. The interpersonal domain is about social interaction, about people dealing with other people. This interaction can be expressed in different ways as given by the LCM. For example, a fight can be described behaviourally (on a physical level) such as [subj] *kicks* [obj] or dispositionally (on a mental level) such as [subj] *despises* [obj].

Job advertisements, however, do not exactly fall into that category of communication. In the texts, the applicant is most of the time the subject and the verbs relate them either to another person or group of persons (e.g. *Je spoort je collega's aan* English: 'You encourage your colleagues'), an action (e.g. *Je presenteert je bevindingen* English: 'You present your findings'), or an object (e.g. *Je brengt de krant rond* English: 'You deliver the newspaper'). This means that not all definitions of the categories as defined by Semin and Fiedler (1991) match precisely with the intent of this task. Therefore, the model had to be adapted to the new domain of job advertisements. Adapting the Linguistic Category Model to the context of job advertisements, the following labels were obtained:

- **Descriptive Action Verb was given the label "Act"**
  DAV was translated to "Act" and described as a single action that can be easily visualized and usually started and completed in a few hours. It is distinguishable with a physically invariant feature.

  Example: *knippen van vlakke platen* English: 'cutting of flat sheets'. Cutting is based on a verb, describing an action with beginning and end, with a physically invariant feature (the action is done by hand). This is the most concrete type of phrasing.

- **Interpretive Action Verb was given the label "Process"**
  IAV was translated to "Process", which is a series of acts or one that can be visualized and/or interpreted in multiple ways. The process is an action that is not distinguished by a physically invariant feature. It has a beginning and end but may take more time (up to days, weeks or months) to complete than an Act.

  Example: *aansturen van vijf medewerkers, werkvoorbereiding / calculatie doorvoeren en inmeten* English: 'managing five employees, carrying out work preparation / entering calculations and measuring'. Managing, entering, and measuring

are all verbs describing actions with no positive or negative valence, with a beginning and end, but without physically invariant feature (managing can be done by pointing/talking/writing, etc.).

Example: *Kortom: je weet klantbehoeftes door te vertalen naar oplossingen en een brug te slaan* English: 'In short: you know how to translate customer needs into solutions and bridge the gap'. To translate and bridge a gap are actions that generally need some amount of interpretation to be understood in context. They are not completely self-explanatory. Translating in this sense is not translation between two languages, and similarly bridging a gap does not mean to physically build a bridge brick by brick. It rather implies a process of finding solutions for problems. Both consist of a combination of more concrete actions.

- **State Verb was given the label "Attitude"**
  SV is called an "Attitude" and should refer to a psychological enduring state, a way of 'being' that is constant over time with a verb as basis. That is, in the context of job ads, a stable way of thinking or feeling. These states cannot be objectively verified.

  Example: *Daarin denk je vanuit concepten* English: 'Therein, you think in concepts'. A way of thinking is not an action but rather a way of 'being' that is stable over time.

  Example: *Je hebt een instelling van wat kan wel i.p.v. wat kan niet* English: 'You have an attitude that looks at what is possible instead of what is not'. This describes a psychological state showing a consequent reaction to being faced with a problem.

- **State Action Verb was given the label "Attitude + action"**
  SAV is called an "Attitude + action" and refers to a psychological enduring state just like a SV, as a result of an action.

  Example: *Je krijgt er energie van op 5 borden tegelijk te schaken* English: 'You get energized from playing chess on 5 boards simultaneously'. Getting energized is a resulting psychological state of performing the action which is playing chess on 5 boards - a metaphor for multitasking.

- **Adjective / Noun / Adverb was given the label "Quality"**
  The label given to the ADJ/NOUN/ADV class is "Quality", because these phrases should describe what the ideal employee is like, thus, what qualities the job advertisement mentions that the person should have. This could be personality traits, skills, or qualifications.

  Example: *Functie eisen: je hebt uitstekende analytische en communicatieve vaardigheden* English: 'Job requirements: you have excellent analytical and communicative skills'. An adjective like "excellent" plus a noun like "skills" that describe someone's stable qualities without specifying what kind of behaviour contributes to this makes that this is the most abstract type of phrasing. Qualities of the company, actions, or objects should not be annotated, as those are irrelevant for the research question.

  "Quality" is further divided into the sub-labels "Innate quality" and "Learned quality". Where Semin only discusses innate qualities like 'honest' and 'impulsive', job advertisements contain many required qualities such as *Je beheerst de Engelse taal* English: 'You master the English language', *Je hebt een rijbewijs* English: 'You have a drivers license', or *Je hebt aantoonbare kennis van Excel* English: 'You have demonstrable knowledge of Excel' which are skills not acquired by nature but

by active learning or training. This is an important distinction to make because the innate qualities can not be validated easily, while the learned ones can be validated with a certificate or test. Besides, the innate qualities tell more about qualities that play a role in the interpersonal domain whereas the learned qualities generally do not.

3.2 Inter Annotator Agreement measures

The Inter Annotator Agreement (IAA) was found by computing Krippendorff's alpha (α) (Hayes & Krippendorff, 2007). This is a reliability measure that can be used for any number of coders. Krippendorff's alpha takes into account stability, reproducibility, and accuracy of the human-annotated scores. Perfect agreement between the raters gives a score of 100%, or 1.0. A score of $\alpha \geq 0.80$ is ideal, and $\alpha \geq 0.70$ is considered acceptable by common standards.

Cohen's Kappa was used as well. It is the most commonly used measure for finding the IAA, and it is also the recommended measure to use according to the writers of the LCM manual (Coenen et al., 2006) for determining reliability between different coders. This measure is, however, limited to two raters. The scores can be interpreted in the same way as Krippendorff's alpha. A Kappa coefficient between 0.61 and 0.80 is seen as high.

3.3 Doccano

The program used for annotation was Doccano (Nakayama et al., 2018), an open source text annotation tool. It provides the ability to present and share various annotation tasks among annotators in a user-friendly way. The annotated documents can be downloaded to be used for tasks like named entity recognition, sentiment analysis, and more.

3.4 Making of the annotation guide

The annotation pilots were only meant to measure the Inter Annotator Agreement for the annotation guide. Therefore, for every annotation pilot, the annotators were presented the exact same small sets of data to annotate.

3.4.1 The final annotation guide

The annotation guide can be found in Appendix **A**

First of all, the annotation guide was written with the goal to label predicates instead of single words. For writing the annotation guide, the LCM Addendum that was used by Johnson-Grey et al. (2020) was helpful for finding the right definitions to describe the different labels for predicates that were based on verbs. Those based on adjectives and exceptional cases were described additionally. The explanations of all labels include 5-10 example sentences from actual job advertisements to demonstrate the application of the LCM in the new domain. To make the labeling more easily comprehensible for annotators, the labels provided were Dutch words that most closely resembled the meaning of each of the categories. Finally, a flowchart was added to help and guide annotators to find the right label for a given predicate more easily.

3.4.2 Data description

For preparing the data for the annotation pilots, an unlabeled dataset of Dutch job advertisements was used. This data was provided by Textmetrics. It consists of 1,190,968 Dutch job advertisements from the year 2014, collected by Job digger. They each have a title text that contains the occupation and a main text that contains the job description. This data was used for running all 5 of the pilots.

Figure 3 gives one example excerpt from the dataset.

Occupation: vertalers

Algemeen Naam van de functie : Gerechtstolk

Functieomschrijving : Beheers jij het Chinese dialect Lei Zhou taal uitstekend in woord en geschrift? Reageer dan snel, wij zoeken voor een omroep in Hilversum een vertaler die voor een programma als tolk kan functioneren. Startdatum is op korte termijn in overleg en voor ongeveer 20 uur. Spreekt deze tijdelijke klus je aan en beheers je de chinese dialect Lei Zhou uitstekend, reageer dan snel. Wij zien je reactie graag tegemoet!

Wijzigingsdatum : 04 mrt 2014
Vervaldatum : 05 mrt 2014

Figure 3: Example job advertisement text.

3.4.3 Data pre-processing

Out of all job advertisements, 194,245 were pre-processed to be prepared to be used in the annotation pilots.

The following was done to clean these texts:

- Special characters were removed using regex such that only the most common punctuation marks were left (“.”, “,”, “?”, “!”, “:”, “;”). In addition, whitespaces were added behind punctuation marks where needed and unnecessary whitespaces were removed.
- The language was detected using the python library ‘langdetect’ (Nakatani, 2014) and non-Dutch advertisements were removed. Out of the pre-processed texts, 183,079 texts were written in Dutch. The second most common language was English, which 10,515 texts were written in. In third place came German with 265 texts. The remainder of the texts were written in more rare cases of various foreign languages.

3.4.4 Conclusions drawn from the annotation pilots

The first four pilots gave unsatisfactory results when computing the IAA. The fifth pilot showed enough agreement amongst the annotators, so the annotation guide used in this pilot was also used for creating the annotated dataset.

All annotation pilots are discussed in detail in Appendix **B**.

From the five annotation pilots, the following conclusions are drawn:

- Labeling predicates is more informative than labeling single words. In the first pilot, annotators had to label single words in the text freely. This gave two problems. First, many relevant words were missed. Verbs like *work*, *do*, or *be* were overlooked easily. Second, the correct label of a verb can vary given the context. For example, in "we are looking for a candidate with a positive attitude", the subject is the company and therefore this part of the sentence is irrelevant, while in "you always look at the positive side", 'look' says something about the candidate and is therefore relevant to annotate.

- It is necessary to define the predicate boundaries beforehand. For untrained annotators, it was shown that it is a tough task to both identify relevant parts of the text they are presented, and give that section of the text the right label. There was doubt about what scope of text should be selected, and many relevant predicates were missed. Therefore, the task had to be turned into a classification task by segmenting the predicates before annotation.

  As a consequence of this extra step, two extra labels are introduced: Incorrect predicate and Irrelevant predicate. The Incorrect predicates are relevant predicates that have been given the wrong predicate boundaries, whereas Irrelevant predicates have been given predicate boundaries but the text inside is not relevant for any of the labels.

- Qualities should be separated as Innate qualities and Learned qualities. In previous pilots, it was specifically asked to label innate person requirements as Qualities. This caused confusion between the categories Attitude and Quality and it also caused some annotators to label learned person requirements as Qualities as well. To make the distinction more clear and the label set more complete, it was decided to add Learned quality as a label, even though it is not based on a category defined in the LCM.

- Attitudes should be separately labeled as Attitude + action when they indicate an attitude related to an action. It was striking that many Attitudes that contained an action were overlooked and only the action mentioned in such phrases was labeled as Act or Process. To counteract this effect, adding the label Attitude + action helped the annotators to correctly identify these cases. This is especially important because the Attitude and Attitude + action classes contain the least samples.

3.5 Making of the labeled dataset

3.5.1 Data description

For creating the final annotated dataset, an unlabeled dataset of 17,810 Dutch job advertisements was used. This data was collected during the second half of 2021 by Textmetrics from a variety of sources such as Job digger and Indeed. The advertisements each have a title text, again containing the occupation, and the main text that contains the job description. This data was used for generating the dataset that is used for further analyses.

Occupation: Job Voorbewerker/ Industrieel spuiterij

In Person Doetinchem - Ulft.
In Person Doetinchem MBO or higher.
Junior/Medior/Senior level.
Permanent contract.
40 hours p/w. MBO or higher.
Junior/Medior/Senior level.
Permanent contract.
40 hours p/w.
Just published.

Over de functie
Jouw werkzaamheden bestaan o.a. uit: Handmatig, hittebestendig spuiten, spatten en natlakken van machines, onderdelen of accenten; Je spuit de machine in de gewenst kleur; Na het spuiten zorg je ervoor dat de gebruikte hulpmiddelen weer netjes schoon worden gemaakt; Je voert af en toe voorbewerkers werkzaamheden uit; Je bent ook verantwoordelijk voor de kwaliteitscontrole van het spuitwerk. Wat wij vragen Voor deze functie vragen wij: - Je bent in bezit van pecifieke cursussen op het gebied van lakspuiten;

- Minimaal 1 jaarwerkervaring als natlakspuiter;
- Je bent kwaliteitsgericht en zeer zorgvuldig ingesteld;
- Je werkt graag in teamverband;
- Je hebt een flexibele instelling en weet van aanpakken.

Wat wij bieden Wij bieden je een fulltime baan voor de langere termijn. Ben jij enthousiast geworden na het lezen van de vacature en pas jij in het profiel? Mocht je vragen hebben dan zijn wij te bereiken op nummer 0314-333901. In Person Doetinchem Terborgseweg 45a 7001GN DOETINCHEM Telefoon: 0314-33 39 01

Figure 4: Example job advertisement text.

Figure 4 provides an example of a job advertisement taken from this dataset.

3.5.2 Data pre-processing

The excerpts, the job advertisement texts, were first extracted and cleaned from HTML tags using a cleaner API from Textmetrics and regex to remove strange characters. This data was sentenized using Spacy ("nl_core_news_sm").

Those advertisements that consisted of less than 5 sentences were removed, leaving 17,764 job advertisements. After removing duplicates, 16,334 job advertisements remained.

3.5.3 Metadata

The data contains the job advertisement texts and also metadata on title, location, and educational requirements. Tables 1, 2, and 3 show some of the top metadata values with at the top the most frequent ones.

Branch	Company
(Uitzendbureaus, 4067)	(ONBEKEND, 778)
(Arbeidsbemiddeling, 1051)	(Tempo-Team, 619)
(Onbekend, 962)	(Randstad, 381)
(Uitleenbureaus, 544)	(Timing, 336)
(Organisatie-adviesbureaus, 276)	(Olympia Uitzendbureau, 273)
(Warenhuizen, 35)	(Synsel Techniek, 220)
(Holdings (geen financiële), 31)	(Luba Uitzendbureau, 208)
(Interieurreiniging van gebouwen, 30)	(Eminent Groep, 199)
(Vervoer over land, 29)	(Actief Werkt! Uitzendbureau, 159)
(Financiële holdings, 27)	(24/7 Chauffeursdiensten, 155)
(Nationale post met universele dienstverplichti...	(WR Werving en Selectie, 150)

Table 1: Most frequent branches and companies present in the dataset

Function title	Profession
(P/W Productiemedewerker, 319)	(Laders en lossers, 1564)
(P/W Magazijnmedewerker, 236)	(Vrachtwagenchauffeurs, 813)
(P/W Logistiek Medewerker, 220)	(Administratief productiepersoneel, 807)
(Operator, 185)	(Monteurs industriële en landbouwmachines, 599)
(Productiemedewerker, 184)	(Informatieverstrekkers, 458)
(P/W Chauffeur, 180)	(Administratieve medewerkers, algemeen, 403)
(Orderpicker, 162)	(Bouwelektriciens e.d., 376)
(Magazijnmedewerker, 157)	(Heftruckbestuurders, 361)
(P/W Administratief Medewerker, 140)	(Verkoopmedewerkers, 349)
(Werkvoorbereider, 140)	(Inpakkers, 329)

Table 2: Most frequent function titles and professions present in the dataset

Location	Province	Education
(AMSTERDAM, 743)	(Noord-Brabant, 3475)	(MBO, 10045)
(ROTTERDAM, 640)	(Zuid-Holland, 3217)	(HBO, 4123)
(UTRECHT, 531)	(Gelderland, 2400)	(VMBO, 2929)
(EINDHOVEN, 489)	(Noord-Holland, 2297)	(WO, 418)
(BREDA, 380)	(Utrecht, 1552)	(LBO, 194)
('S-GRAVENHAGE, 367)	(Overijssel, 1369)	(GEEN, 53)
('S-HERTOGENBOSCH, 305)	(Limburg, 1102)	(HAVO, 30)
(ZWOLLE, 304)	(Friesland, 590)	(ONBEKEND, 18)
(APELDOORN, 297)	(Groningen, 549)	
(GRONINGEN, 296)	(Drenthe, 455)	
(TILBURG, 289)	(Zeeland, 415)	
(ARNHEM, 232)	(Flevoland, 389)	

Table 3: Most frequent locations, provinces, and educational backgrounds present in the dataset

3.5.4 Sampling strategy

Because the dataset is imbalanced considering its metadata, as was seen in section 3.5.3, a sampling strategy was used.

It was decided to first divide the samples using the variable 'Branch'. This strategy gives a set of sentences half of which are from employment agencies. The reason for this is that employment agencies mean to help people get a job, and those people generally find it challenging to find a job by themselves. They offer entry-level jobs, be it temporary or long term work. They help people enter the job market, and thus need to take into account both the wishes of the employer and the employee. Therefore, it is especially important for those job advertisements to be formulated in an accessible way.

Then, the samples of both sets were balanced further over the variables company and profession from the metadata. These variables show that some specific companies and jobs are over-represented, in specific "Tempo-Team" for company, and *laders en lossers* English: 'loaders and unloaders' and *vrachtwagenchauffeurs* English: 'truck drivers' for profession. Randomizing over the company or job type should make sure that the final annotation sentences are pulled from data with a fairer representation of the distribution of jobs than if they were randomly sampled from the original set.

The dataset was divided into two sets:

- The first set contains jobs in the branch of employment agencies, in specific those labeled as *uitzendbureaus* English: 'employment agencies', *uitleenbureaus* English: 'lending agencies', *arbeidsbemiddeling* English: 'job placement', or *organisatie-adviesbureaus* English: 'organizational consultancy firms'. For sampling from the first set, a company was selected 3,000 times at random and then a post of that company was selected, and then a random sentence from that post was picked. The random sentence was the sentence in the middle of the post with a deviation of a random number in the range [-2, -1, 0, 1, 2]. This sentence was saved, if it was over 30 characters long, was not already in the set of saved sentences, if it did not contain information like a phone number or decimal, and if it did not end with a ":", because those sentences are generally not relevant for annotation. This post was then dropped from the dataset and the next sample was taken.
- The second set contains all 'other' ads that are not of employment agencies. For sampling from the second set, 3,000 sentences were found in the same way as above, except that they were picked at random from a job type rather than from a company.

This provided random sentences from the distribution of job posts as given in table 4.

Companies, employment agencies (3000 posts)	Professions, not employment agencies(3000 posts)
(Walters People, 27)	(Vertegenwoordigers, 69)
(IQ Select, 26)	(Systeembeheerders, 61)
(Aethon Publica, 26)	(Informatieverstrekkers, 60)
(WR Werving en Selectie, 26)	(Administratief productiepersoneel, 58)
(Derec, 25)	(Administratieve medewerkers, algemeen, 55)
(Abiant Personeelsdiensten, 25)	(Monteurs industriële en landbouwmachines, 51)
(Tosca Medisch Interim, 25)	(Tuinders en kwekers, 51)
(Xelvin, 25)	(Vakspecialisten op het gebied van maatschappel...
(Yacht, 24)	(Specialisten op het gebied van reclame en mark...
(Matchpartner, 24)	(Bouwelektriciens e.d., 48)
(Tempo-Team, 23)	(Kelners, 48)
(Arto Uitzendbureau, 22)	(Applicatieprogrammeurs, 47)
(Artiflex, 22)	(Accountants, 47)
(V-NOM, 22)	(Laders en lossers, 46)
(Talent&Pro, 22)	(Televerkopers, 45)

Table 4: On the left: the most frequent companies when sampled sampled according to the sampling strategy from the set of advertisements that come from employment agencies. On the right: the most frequent professions when sampled according to the sampling strategy from the set of advertisement that do not come from an employment agency.

3.5.5 Predicate segmentation

For determining the predicate boundaries, similar steps were followed as were introduced in pilot 5. These are explained in appendix **B.5.0.1**.

The sentences were parsed by Frog (van den Bosch et al., 2007). Using the following rules, the predicate boundaries were defined automatically:

- All predicates are complete verb phrases so they usually start with a word with tag “B-VP”.
- Sometimes, they start with the (in)finitive form of a verb, so look for:

  “WW(inf,nom,zonder,zonder-n)”

  “WW(inf,vrij,zonder)”

  “WW(pv”
- Sometimes, they start with an adjective, so also search for:

  ”ADJ(prenom,basis,met-e,stan)”

  and [”ADJ”, ”vrij”, ”basis”, ”zonder”]
- Some predicates start with specific relevant nouns: [“ervaring”, “kennis”, “beheersing”, “affiniteit”, “interesse”, “gemotiveerd”, “big”, “verantwoordelijk”, “zelfstandig”, “mbo”, “hbo”, “wo”, “vmbo”, “lbo”, “havo” “niveau”, “technisch”, “inzicht”, “opleiding”, “diploma”, “certificaat”, “bezit”]
- End the predicate when encountering a punctuation mark “.”, “!”, “?”, “;”, or “:”.

- Sometimes end the phrase at “en” or “,”, namely only if a new predicate beginning is found after the phrase, or if the dependency level for the word after the comma is lower than for the word in front of the comma. In that case, it likely belongs to a new verb phrase that is introduced later. In enumerations, the words usually have the same relation to the main verb so they are on the same dependency level. These are added to the same predicate.

This resulted in predicates as given in figure 5.

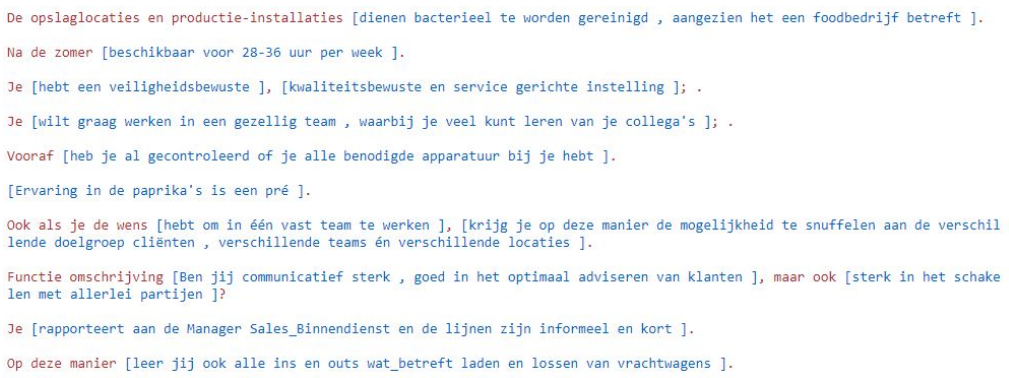


Figure 5: Random selection of sentences with the automatically found predicate boundaries given in blue between square brackets.

As mentioned before, many of the sentences are not grammatical. This was not corrected for in any way. Therefore, the parsing is not completely stable. For example, some verbs were given a noun label and therefore the verb phrase was not given the right boundaries. To make up for these cases, the list of extra words was added manually as defining the start of a new predicate. This is a big disadvantage of segmenting the predicate beforehand. It is a problem that is hard to correct for automatically and it is also too time-consuming to correct thousands of sentences by hand.

3.5.6 Inter Annotator Agreement scores of the final dataset

For obtaining the final annotations, three annotators annotated the set of sentences. They each labeled respectively 780, 893, and 1230 automatically segmented predicates. The three sets had an overlap of 100 sentences (139 predicates) to find the Inter-Annotator Agreement.

- Comparing the annotations on word level, including non-relevant parts of the texts, a Krippendorff's alpha of 0.87 was obtained.
- Comparing the annotations on predicate level gave a Krippendorff's alpha of 0.77.

On predicate level, the confusion tables given in figures 6, 7, and 8 were found.

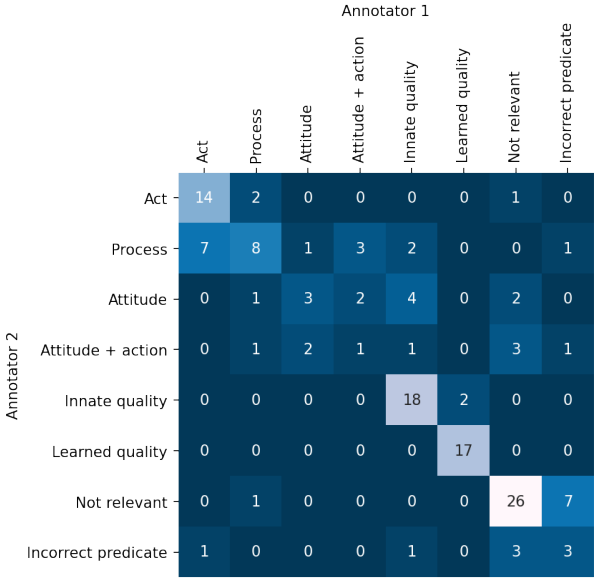


Figure 6: Confusion table of all labels between annotator 1 and 2

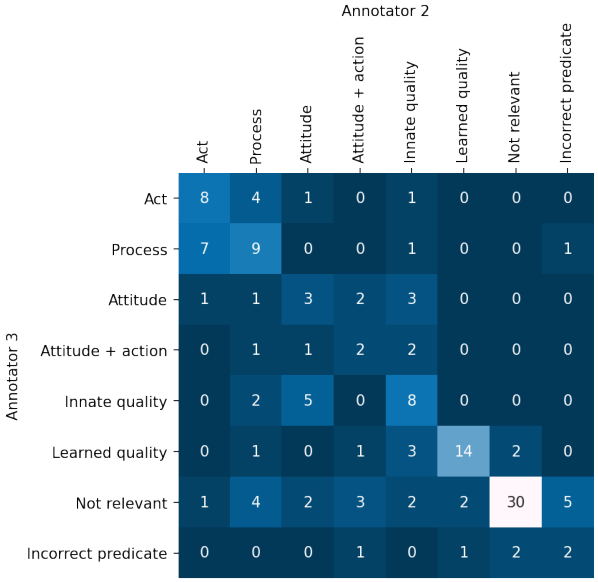


Figure 7: Confusion table of all labels between annotator 2 and 3

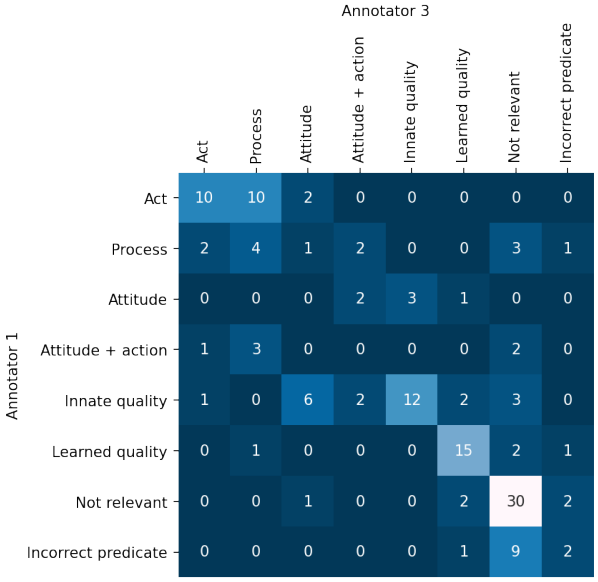


Figure 8: Confusion table of all labels between annotator 3 and 1

Other than in the pilots, the sentences were selected according to the sampling strategy as explained in 3.5.4. In the pilots, the sentences were selected by hand, based on relevancy. The tables show that the annotators labeled respectively 35, 49, and 34 predicates as not relevant and 12, 8, and 6 as having incorrect predicate boundaries. The incorrect boundaries may or may not contain relevant information, but because that is not given, those are left out. In total, 139 predicates were provided. That means that, on average, 39 of all found predicates can be left out because they are not relevant and 9 can be left out because their boundaries are incorrect, making up respectively 28% and 6% of all predicates, and together 34%. Almost one third of the annotations can be dropped after annotation, if you were to keep only relevant and correct predicates.

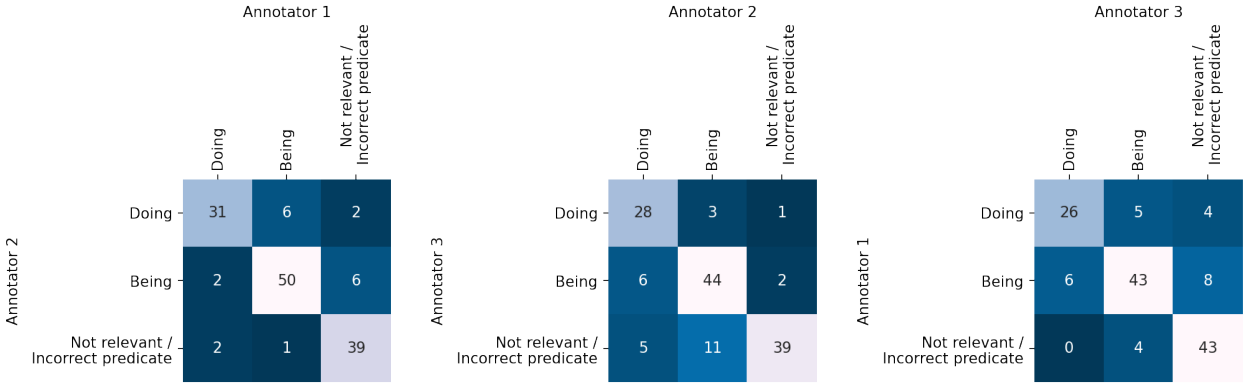


Figure 9: Confusion tables between all annotators. “Being” contains predicates labeled as Attitude or Quality, and “Doing” contains predicates labeled as Act or Process.

Separating the predicates on the basis of “Doing”, “Being”, and “Not relevant/Incorrect” gives a Krippendorff’s alpha on predicate level of 0.76. Figure 9 shows this.

4 Data description

This chapter shows the class distribution of all annotated predicates in the final dataset that was used to build on in the next chapters. It explains how the dataset was split into separate training, validation, and test sets, and define baseline scores for classification.

4.1 Class distribution

In total, 4,000 sentences were annotated. These consisted of 5,227 predicates. 262 of these were labeled as having incorrect predicate boundaries. These are not suitable for training a model so they are discarded, leaving 4,965 predicates with correct predicate boundaries.

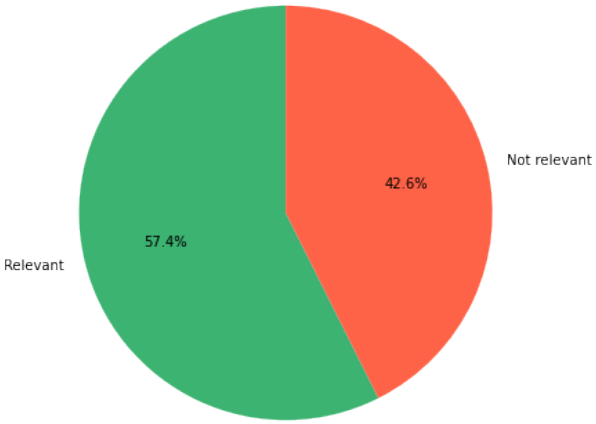


Figure 10: Distribution of the relevant and non-relevant correct predicates.

As is evident from figure 10, more than half of the predicates are relevant for this topic. These predicates are either about the candidate or about job-related activities. The relevant labels are investigated further.

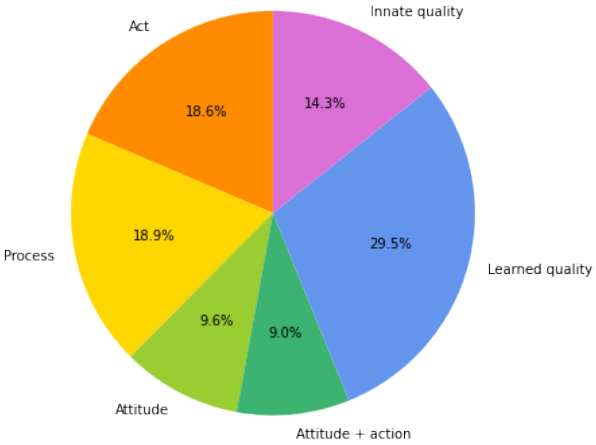


Figure 11: Distribution of predicates with correct and relevant (sub-)labels.

Figure 11 shows that the Learned qualities label is slightly over-represented. The Attitude + action and Attitude labels are under-represented, however they could also be jointly considered as one class (like the SVs and SAVs of the LCM).

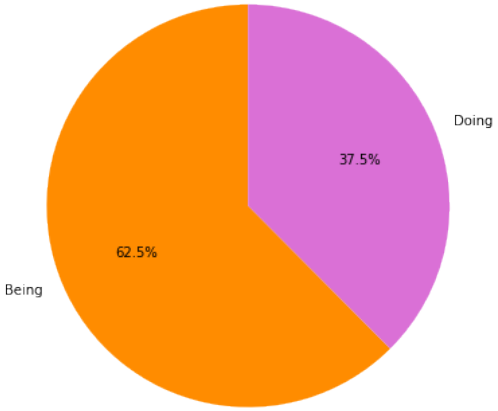


Figure 12: Distribution of predicates with correct and relevant labels, simplified.

'Being' contains all dispositional phrases. That is, all Attitudes and Qualities, as defined as S(A)V and ADJ in the LCM. These are person requirements asked of the applicants. 'Doing' contains the behavioural phrases, so that includes the Acts and Processes or the DAVs and IAVs as defined in the LCM. They are descriptions of the work-related tasks and processes. Figure 12 shows that there are more dispositional predicates than behavioural predicates in the dataset.

4.2 Training/validation/test split

The dataset was split using a manually implemented stratification strategy such that the labels are equally and proportionally distributed over the sets of predicates. Table 9 shows how the predicates and their corresponding labels are distributed over the training, validation, and test sets.

	Training	**Validation**	**Test**	**Total**
Act	370	80	80	**530**
Process	378	81	81	**540**
Attitude + action	190	42	41	**257**
Attitude	179	39	39	**273**
Innate quality	285	61	62	**408**
Learned quality	589	127	126	**843**
Being	1,243	269	268	**1,780**
Doing	748	161	161	**1,070**
Not relevant	1,480	318	317	**2,115**
Incorrect window	183	40	39	**262**
Total	**3,654**	**788**	**785**	**5,227**

Table 5: Class distribution of the training, validation, and test sets over all predicates.

This gives a stratified train-val-test split of the dataset as follows:

- Training set: 70% = 3,654 predicate (2,791 sentences)
- Validation set: 15% = 788 predicates (623 sentences)
- Test set: 15% = 785 predicates (587 sentences)

4.3 Defining the baseline

Multiple models are going to be compared to show how they perform on a sequence labeling task. To be able to state how well they perform, two baseline scores were calculated for each subdivision of the task. They show the accuracy for random labeling, where every sample is given a random label from the dataset, as well as the most frequent label, where every sample is given the label that occurs most frequently in the dataset.

1. Task: sequence classification

   Relevant vs. non-relevant

Baseline type	**Accuracy**
Uniform	0.48
Most frequent	0.58

Table 6: Baseline scores for classifying by relevancy.

Dispositional ('being') vs. Behavioural ('doing'):

Baseline type	**Accuracy**
Uniform	0.54
Most frequent	0.61

Table 7: Baseline scores for classifying by dispositional/behavioural label.

All relevant labels:

Baseline type	**Accuracy**
Uniform	0.19
Most frequent	0.29

Table 8: Baseline scores for classifying by LCM subcategory label.

2. Task: sequence tagging (word level classification with majority class label being "O" given to all irrelevant words)

Baseline type	**Accuracy**
Uniform	0.13
Most frequent	0.63

Table 9: Baseline scores for classifying on word level including the label "O".

5 Approach

The original approach of automating the LCM is done with a dictionary mapping specific verbs to classes. These words were found through analyzing manual annotations. This would be a rule-based approach: if [word] is in dictionary, then give it the corresponding label. However, there is a disadvantage to this type of system.

It is only possible to create this dictionary for problems that are clearly defined. That is, if a (set of) words always, without exception, belongs to one category. However, since we are looking at predicates, this is expected not to be possible. In one predicate, a word can be indicative for a different labeling than in another sentence. For example, take the word "responsibility". This word can occur in a sentence such as "You are responsible" or "You have a great sense of responsibility", in which case it would indicate a quality. It could also occur in sentences such as "You are responsible for [action]" or "We are not responsible for your mistakes" in which case it should not be labeled as quality.

Because of this limitation of a dictionary-based system, we use supervised machine learning to make the system generate and learn the rules on its own from the training examples. This section will explain which supervised models were implemented as well as how they are expected to behave in specific to the provided data and task.

5.1 Sequence classification

Sequence classification makes up two steps of the three-step sequence tagging approach. After detecting the predicate boundaries, the predicates are classified by relevance and LCM categories. The classification models used for this are built as described in this section.

5.1.1 Machine learning algorithms

The first set of models applied consists of Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, and XGBoost. These models were implemented with scikit-learn (Pedregosa et al., 2011), the most popular library for machine learning in Python, especially for supervised classification tasks. These methods are surpassed in performance with the arrival of word embeddings and transformers, but they nonetheless are useful and faster to run. They are also flexible, having the option of feature engineering and hyperparameter tuning, and thus can show like no other model today which features are the most helpful for identifying different classes. The data is first cleaned (pre-processed), then the most important features are extracted, before they are fed to the model and the model's performance is evaluated.

5.1.1.1 Data pre-processing

The predicates were:

- Tokenized in NLTK (Bird et al., 2009)
- Lemmatized using Spacy (Honnibal & Montani, 2017)
- Unknown (UNK) labels were removed
- Punctuation marks were removed

- Stop words were removed, except for “zijn”, “doen”, “geweest”, “moet”, “kan”, “kunnen”, “wil”, “worden”, “wordt”, and “wezen”. These stopwords are kept in the data as they are expected to be important features for the category Attitude + action.
- Texts smaller than one word were removed

5.1.1.2 Feature extraction and representation

To be able to further analyze the pre-processed texts with a classification algorithm, the texts need to be converted to a numerical format. This was done with vectorization. Both the inverse document frequency weightings over uni-, bi-, tri-, and quadrigrams were extracted from the training data, as well as the word count (BoW) weightings. Both were set to have a minimal occurrence of the n-gram in the data set of 2, to account for significant occurrence of each word.

Dimensionality reduction

Although the vectorized representations also account for the entire corpus that the model knows, not all words are useful for predicting the classifications. When a feature occurs in most to all texts, it loses meaning. On the other hand, the most infrequent words can add noise and therefore make it harder for the model to generalize. Dimensionality reduction was done as a process of trial and error. To remove sparse features from the data but still ensure that every predicate at least contained some relevant words features for classification, the dimensionality was reduced to no less than 500 features.

The dimensionality of the vectors was first reduced by variance thresholding. The TF-IDF vectors started with a dimensionality of 1,833. With variance thresholding (threshold=0.0005) this was reduced to 573 features and with the Chi-Square test this was further reduced to vectors of length 500.

The features with top tf-idf scores are: (‘zijn’, 0.032), (‘ervaring’, 0.027), (‘werken’, 0.015), (‘goed’, 0.015), (‘kennis’, 0.014), (‘jij’, 0.013), (‘opleiding’, 0.010), (‘gaan’, 0.010), (‘klant’, 0.010), (‘werkervaring’, 0.010))

The Bag of Words vectors also started with a dimensionality of 1,833, which was reduced to 681 by variance thresholding (threshold = 0.002) and also further reduced with the Chi-Square test to vectors of length 500.

The features with top word count scores are: (‘zijn’, 279), (‘ervaring’, 162), (‘goed’, 117), (‘jij’, 96), (‘werken’, 95), (‘werk’, 78), (‘opleiding’, 75), (‘kennis’, 74), (‘klant’, 66), (‘werkervaring’, 64)

These vectorized sets of 500 features had been fit on the training data and were later applied to the validation and test data to convert them to the same format. Nearly all n-grams of size bigger than 1 were too sparse and therefore removed in this dimensionality reduction step.

For example:
Predicate text: kennis van pedagogiek en ontwikkelingspsychologie is goed ontwikkeld
Words in vectorizer: [‘goed’ ‘kennis’ ‘ontwikkelen’]
TF-IDF scores for the words in vectorizer: [0.45699751 0.58593967 0.6691995]
Vectorized representation: [0, 0, ..., 1, ..., 1, ..., 1, 0, 0, 0]

5.1.1.3 Models used

Decision Tree (DT)

The first method is a Decision Tree. This approach is most in line with a rule-based system, since there are rules/conditions involved in following the path from root to leaf node. Applying this method consists of two stages: 1.) constructing a tree and 2.) pruning the tree. From the root node, the Decision Tree algorithm divides the data into separate sets based on one feature (the feature that gives most information gain or least impurity as measured by entropy). The child nodes of the root node, again, split into two separate sets in similar fashion and this continues until the nodes cannot split any further or a set maximum depth is reached. There is often a limit set to the depth of the tree to avoid overfitting and there are also other choices made for what feature to split on, such as a minimum number of samples per leaf node. These parameters are useful for smoothing.

The Decision Tree makes sense to apply to this problem because of its explainability. The decision to classify a predicate as one or another is made based on the specific combination of the words it contains. Fitting a Decision Tree to the - either BoW or TF-IDF - vectors can show which word combinations cause the tree to split at various points. Because the problem is expected to be more complex than a Decision Tree can capture, this model is expected to be likely to underfit.

Random Forest (RF)

The random forest is an integration-based method that is meant to boost the performance of the single Decisions Tree by constructing many subsets of the data and building Decisions Trees (or estimators) for each of them. Then, for a new sample, it can run it through every tree and calculate the majority label of the labels that result from each of the sub-trees. This method of using multiple trees and choosing the majority label is called 'bagging', and it can capture more complex dependencies and gives better generalized results than a single Decision Tree.

Gradient Boosting (XGB)

Gradient boosting is another method based on Decision Trees. Where a Random Forest processes many Decision Trees in parallel, a Gradient Boosting algorithm generates a new Decision Tree at each step, and builds a new, 'better' tree, based on the residuals (errors) that resulted from the previous tree. It learns to improve itself and fit the samples it is provided better at each step. This model is however more prone to overfitting. This can be counteracted by choosing a low learning rate, such that it only slowly adjusts itself to the training data.

Support Vector Machine (SVM)

The SVM model takes the vectors of the samples and projects them as nodes in a multidimensional space. It draws a hyperplane through this space for every binary classification decision, maximizing the distance between the hyperplane and the nodes of those two categories. The category boundary will then be drawn such that its distance along the direction perpendicular to the hyperplane is the largest. The kernel function is of importance for ensuring that the model can become a robust nonlinear classifier. A linear kernel was used for this task.

Naïve Bayes (NB)

Naïve Bayes is the simplest type of probabilistic graphical models. It is broadly used in text classification tasks because of its simplicity. It creates a probability distribution for the class labels $p(y)$ and a probability distribution of features over class labels $p(x_i|y)$. It then primarily uses the prior probability of a class label to calculate the posterior probability of the class label given the features, with the assumption that all features are independent of each other (hence "naïve"). The features are treated as being independent given the target label, even though they might in fact not be independent.

5.1.2 Word2Vec

For Word2Vec, the pre-trained Dutch embeddings used were taken from a repository as described by Tulkens et al. (2016). This pre-trained dataset has a vocabulary size of 1,442,950 and contains 320-dimensional word embeddings. The data was trained on a combination of sources: Roularta, Wikipedia, and Sonar500. These pre-trained embeddings are suitable to use as input to a neural network.

5.1.2.1 Data preprocessing

The data was converted to lowercase and stripped of punctuation marks. The stopwords were left in because they can be helpful for the LSTM to contextualize the relevant words.

5.1.2.2 Vector representation

The preprocessed sentences were converted to numeric vectors by looking up the words in the word embeddings and padding them to a maximum length of 20.

For example:
Input sentence: heb jij aantoonbare affiniteit met of ervaring in de zorg
Vectorized sentence: [34 21 154 85 8 13 15 6 2 79 0 0 0 0 0 0 0 0 0 0]

5.1.2.3 The network architecture

On top of the embeddings, an LSTM layer was added with dropout and recurrent dropout to combat overfitting and a dense layer with softmax activation function was added for finding the label probabilities. The network's architecture is shown in figure 13.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 20, 320)           1061760
_________________________________________________________________
lstm (LSTM)                  (None, 64)                98560
_________________________________________________________________
dense (Dense)                (None, 6)                 390
=================================================================
Total params: 1,160,710
Trainable params: 98,950
Non-trainable params: 1,061,760
_________________________________________________________________
```

Figure 13: Model summary for Word2Vec classification with an LSTM layer. The binary tasks had a dense softmax output layer for 2 classes.

5.1.2.4 LSTM

LSTMs are RNNs with some extra capabilities to learn long-term dependencies in text. They are widely used for various types of tasks. It is different from an RNN because it has a cell state that is meant to remember information while moving further along in the text, now and then allowing new information to be passed to the cell state through gates. This way, it is better at learning what information in the input is relevant to keep and what is not.

5.1.2.5 Visualizing Word2Vec

To get an idea of the initial layout of the Word2Vec word embeddings in relation to the labels, a scatter plot was generated to show several word embeddings of important features in relation to each other, as transformed to 2D space with t-SNE. This is shown in figure 14.

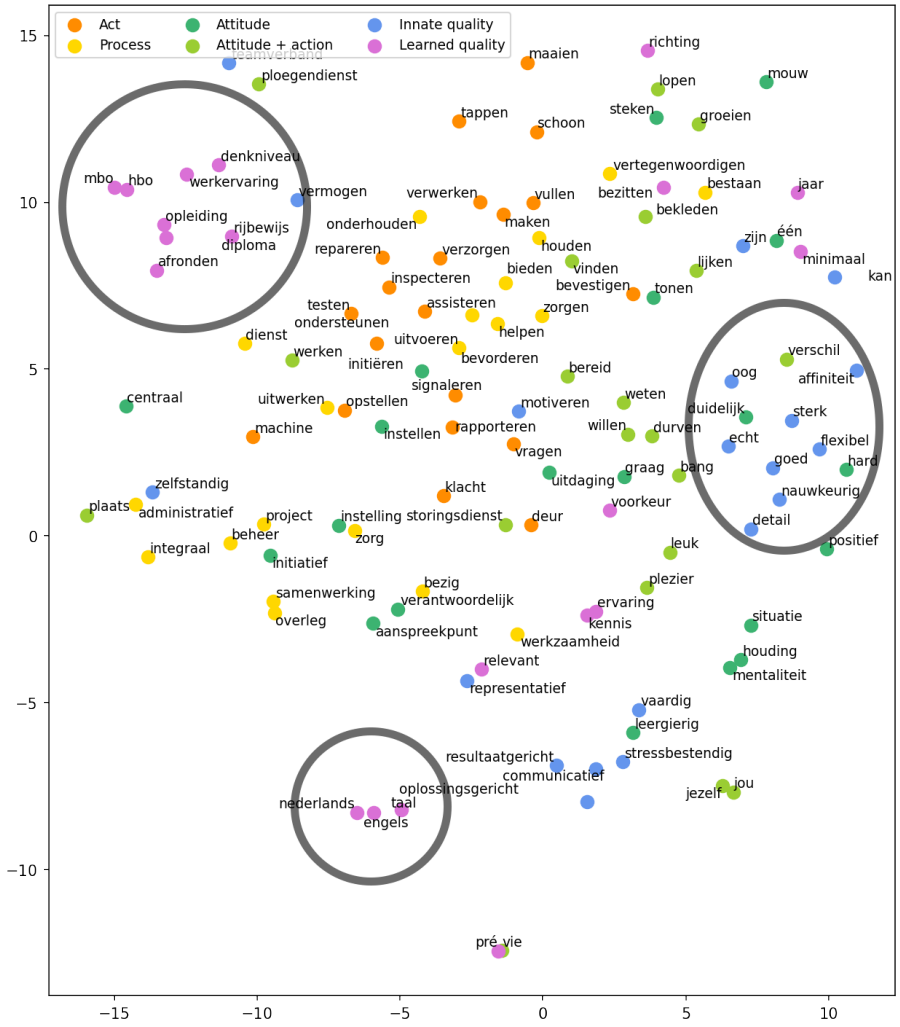


Figure 14: Word2Vec embeddings of top features (those that have the highest average TF-IDF scores for every class), visualized by t-SNE with a perplexity of 20.

Data visualisation with t-SNE

T-SNE is a method to visualize multi-dimensional information in a 2D or 3D plot. The conversion from Word2Vec space to 2D space was made using t-Distributed Stochastic Neighbor Embedding, a technique introduced by Maaten and Hinton (2008) that is used for dimensionality reduction. It is a commonly used technique to reduce the dimensionality of word embedding vectors, as an alternative to Principle Component Analysis. The Word2Vec word embeddings originally are 320-dimensional vectors and after applying t-SNE this was reduced to 2-dimensions. Much of the high-dimensional information was preserved. The most important tuneable parameter for t-SNE is perplexity. This is a number that should be an estimate of the number of close neighbors that each datapoint has (Wattenberg et al., 2016).

Interpretation of the t-SNE plot

The features shown in figure 14 are the top 20 selected features per class as were selected by the TF-IDF vectorizer. For every label, they are the features that have the highest average TF-IDF scores. The perplexity chosen to create this plot was 20, because the input given was 20 features for each of the classes. After analyzing multiple plots, varying the perplexity between 10 and 40, 20 seemed still to give the best clustering of the classes. The number of steps to create this plot was 3,000.

It is hypothesized that using pre-trained Word2Vec embeddings has advantages over using TF-IDF vector representations. First of all, TF-IDF vectors represent every word with one number, while Word2Vec has 320 dimensional vectors for every word. This means that words can be related to each other in a multi-dimensional vector space. Another advantage is that Word2Vec vectors can be pre-trained. Because of the relatively small dataset used in this project, out of vocabulary words can therefore also be recognized and related to their context.

T-SNE internally uses both local and global adjustments, which makes it hard to interpret how the dimensions are reduced, but the plot shows that t-SNE tries to find clusters of features. The most remarkable cluster is for Learned qualities, which seem to stand somewhat alone in the top left corner. All of those terms seem to be related to education. There is a small cluster at the bottom for languages. The center right shows many Innate qualities.

There is much overlap between the features of other classes. The center top shows another important group of words, as this area contains mostly verbs that are important for Acts and Processes. Still, the t-SNE does not show clear clusters for most classes. What this suggests that the classes are not easily separable using only Word2Vec embeddings (with the exception of Innate and Learned qualities). Context given by the other features in the predicates might be important to identify the right classification.

5.1.3 Transformers

For the task of predicate classification, two transformer models were fine-tuned: BERT and RoBERTa.

5.1.3.1 BERT

BERT stands for Bidirectional Encoder Representations from Transformers (Devlin et al., 2018). It is a pre-trained language model that has a general understanding of a

specific language or multiple languages at once. The information captured is saved in the form of WordPiece embeddings. Every WordPiece is embedded by 769 real numbers, which represent their semantic relation in a feature dimension to every other WordPiece in the corpus.

BERT was pre-trained in unsupervised way with a masked language model objective. For this task, random words in an input text are masked and the BERT encoder learns to predict the masked words from the context of the text. Self-attention is the idea that the relations between every possible combination of words in the input are examined, such that both the text before and after the masked words are taken into account. This is also meant with the bi-directionality of the model.

BERT's corpus is tokenized in such a way that words that consist of common structures are split up into WordPieces or sub-words. For example, look at the word "em ##bed ##ding". "bed" and "ding" are not the beginning of the words (denoted by ##) and they are also common parts to come back in other words so they have their own embeddings. Sub-words form a compromise of word embeddings and character embeddings. The difference in use is that, for word embeddings, usually a separate "OOV" vector is added to the corpus for those words that are not known, that are "out-of-vocabulary". ELMo is a transformer model that tries to solve this issue with character embeddings. These have been used extensively to be able to also embed tokens that were never seen during training, but still get a different embedding for each new string of characters found. The WordPiece representation that BERT makes use of allows for unknown words to be at least partly identified.

Originally, RNNs and later LSTMs were a popular way of capturing long-distance dependencies within texts. Nowadays, the attention principle fixes this more efficiently. The first Transformer architecture stacked encoders on top of each other. The encoders mapped tokens to vectors that capture semantic and linguistic constructs of the language better every time. Each encoder has an attention mechanism and a feed forward neural network. The feed forward neural network then has the task to normalize the vectors such that they can be fed to the next encoder for further processing. This stack of encoders then is the pre-trained part of the Transformer. This can be led through to the decoders, the mechanism that interprets the trained embeddings and returns language based on them and the task. The decoders are missing from the downloadable BERT models because that is the part that needs to be trained on top of the embeddings before use. The encoders provide all necessary general language information needed for this.

The two models used most often are bert-base and bert-large. Bert-base has 12 encoders, while bert-large has 24. It can be imagined that bert-large, therefore, has a slightly better tuned knowledge representation of the language. This also does show in its performance generally. In this project, however, bert-base was used for efficiency.

The network architecture

The pre-trained transformer model ("GroNLP/bert-base-dutch-cased") was fine-tuned for the sequence labeling task. The network's architecture is shown in figure 15.

An extra dropout layer was added for regularization.

```
Model: "model"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_ids (InputLayer)       [(None, 30)]              0
_________________________________________________________________________________________________
attention_mask (InputLayer)  [(None, 30)]              0
_________________________________________________________________________________________________
tf_bert_for_sequence_classifica TFSequenceClassifier 109142022   input_ids[0][0]
                                                                 attention_mask[0][0]
_________________________________________________________________________________________________
dropout_38 (Dropout)         (None, 6)                 0           tf_bert_for_sequence_classificati
_________________________________________________________________________________________________
dense (Dense)                (None, 6)                 42          dropout_38[0][0]
=================================================================================================
Total params: 109,142,064
Trainable params: 109,142,064
Non-trainable params: 0
_________________________________________________________________________________________________
```

Figure 15: Model summary for BERT classification. This is the network for classifying all sub-labels. The binary tasks had a dense softmax output layer for 2 classes. For RoBERTa, the tf_bert_for_sequence_classification was changed to tf_roberta_for_sequence_classification.

BERT/BERTje tokenizer example

Tokenized test: [[CLS], heeft, een, pro, ##actie, ##ve, werk, ##houding, [SEP], [PAD], [PAD], ..., [PAD]]
Input ids: [1 13117 11130 17752 23827 29317 22381 25998 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3]
Attention mask: [1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Label: Attitude

The tokenized sentence input to BERT starts with a [CLS] token. For sentence-level classification, this token will, when trained, contain a representation in the form of an embedding (768-dimensional) of the full input predicate. BERT recognizes new texts by the [SEP] token, which provides the information to the model that the next sentence is not a direct follow-up of the current one. The number 3 in input ids stands for the padding tokens which are added because all inputs need to have the same length.

5.1.3.2 RoBERTa

RoBERTa (Liu et al., 2019) was later introduced by Facebook as an improved model of BERT. RoBERTa also used masked language modeling, but along with that, it was trained on a next sentence prediction (NSP) task. In this task, the input is two sentences and BERT has to decide if the second sentence is a random sentence or if it was the original follow-up of the first sentence in the corpus. By trying to optimize for this objective, the performance is boosted for finding long-distance relationships in texts. Its performance was further improved by tuning BERTs hyper-parameters. RoBERTa was also pre-trained on a bigger dataset than BERT.

The network architecture

The pre-trained RoBERTa model ("pdelobelle/robbert-v2-dutch-base") was used to fine-tune for the sequence labeling task. The network architecture is the same as was used for BERT.

RoBERTa/RobBERT tokenizer example

Tokenized text: [<s >, is, Ġloy, aal, ',', Ġbetrouwbaar, Ġen, Ġst, res, st, oler, ant, </s >, <pad>, ..., <pad>]
Input ids: [0 372 27384 718 6 4877 8 691 2193 273 24018 550 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
Attention mask: [1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Label: Innate quality

5.1.4 Evaluation metrics

Sequence labeling is a typical text classification task. Therefore, the following measures were used to evaluate the performance of the models:

- Accuracy:
$$\frac{TP + TN}{TP + TN + FP + FN}$$
- Precision:
$$\frac{TP}{TP + FP}$$
- Recall:
$$\frac{TP}{TP + FN}$$
- F1-score:
$$\frac{2 * Precision * Recall}{Precision + Recall}$$

Since the tasks are not intended to find positive and negative values but rather to find the correct label for predicates, each task is treated as multi-class task with either 2 or 6 classes. The classes are equally important to detect, therefore, F1-score is supposed to provide the most meaningful scores out of those shown above. It balances the precision and recall. The F1-score is also generally treated as being the best measure to compare against the Inter-Annotator Agreement which then functions as the upper-bound performance that can be achieved by a model based on that same data. There are studies suggesting that a model can 'outperform' the IAA (Boguslav & Cohen, 2017), however this is unlikely to happen because of the complexity of understanding the meaning of predicates. The model would need to be more knowledgeable than the annotators, or it would need to learn the bias of every annotator. The information necessary for that was dropped from the data set.

The Receiver Operating Characteristic curves and Precision-Recall curves were also obtained for all problems. This was found by plotting the one-versus-rest curves, which then represent the model's ability to distinguish between one class in specific (seen as the positive class) and the other classes joined together. The sensitivity and specificity are reflected in these ROC curves. The mean Area Under the Curve was found for both types of curves by averaging them over the AUC's found for every label.

5.1.5 LIME for interpretability

LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016) is a model-agnostic tool that can attempt to explain the predictions of any classifier. It learns a model around the predictions made by approximating them locally. From the samples it is presented, it gives a set of perturbed samples, where each perturbed sample

contains a random fraction of the original sample. By running many perturbed samples through the model, a locally faithful explanation is found. 'Local', in LIME, refers to 'local fidelity', which means that the model should be able to replicate the results of the classifier for samples in the neighborhood. LIME is applied to get insight in which parts of the sample predicate were most important for deciding on the final classification. This is very helpful, especially when using black-box models like Transformers. LIME can be used to explain why one model behaves differently than another.

5.2 Sequence tagging

This section explains how the one-step sequence tagging approach was set up. The model presented, is meant to tag each token directly as belonging to an LCM category or not, such that the task is done in one step. The predicates are derived from these token tags, so the process is, in a way, reversed, compared to the three-step sequence tagging.

5.2.1 Transformers

There are many ways to implement sequence tagging. In this thesis, the BERT transformer model was used. It is most comparable to the BERT classifier that was used in the three-step approach.

5.2.1.1 BIO tagging scheme

For processing the annotations, the tagging scheme called 'BIO' (also known as 'IOB') was used. This is the most popular tagging scheme for named entity recognition tasks. This scheme is also suitable to apply to the data in this project because the approach is very similar. BIO stands for BEGIN-INSIDE-OUTSIDE. The list of possible tags consists of all combinations of the labels ("Act", "Process", "Attitude + action", "Attitude", "Innate quality", "Learned quality") and prefix ("B-" or "I-") and a separate label "O", which is the label for tokens that are not part of an annotated predicate.

5.2.1.2 Data preparation

The sentences were:

- Tokenized (with a maximum length of 80 tokens). Tokenization is the first step to be able to tag the separate tokens of the sentences.
- The tokens were given corresponding BIO labels. The first token of every annotated predicate was given the "B-" as prefix to the corresponding label, the other tokens in the predicate got the "I-" prefix, and the tokens outside of the predicates in the sentences were given the label "O".

5.2.1.3 The network architecture

The pre-trained transformer model ("GroNLP/bert-base-dutch-cased") was fine-tuned for the sequence tagging task. The network's architecture is shown in figure 16.

5.2.2 Evaluation metrics

Sequence tagging tasks can not be evaluated with the same metrics as sequence classification tasks, because the predicate boundaries are not guaranteed to be the same. Therefore, those metrics are only applicable for evaluation on word level. However, there

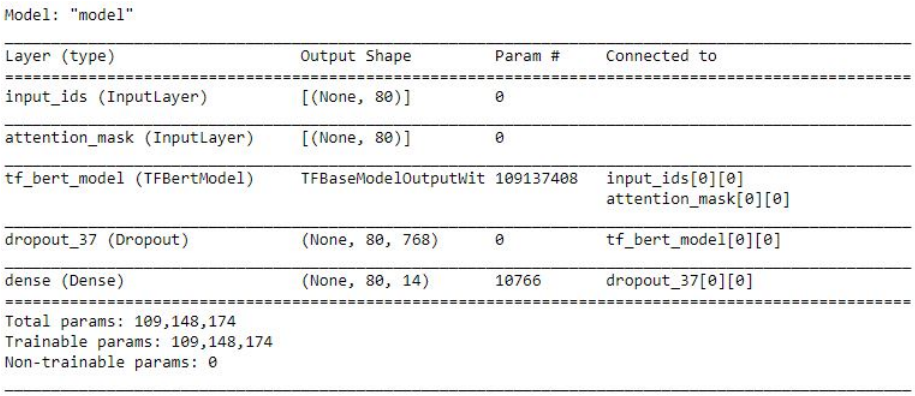


Figure 16: Model summary for BERT sequence tokenization. The 14 output classes stand for "O", 6 sub-labels * 2 for "B-" and "I-" tag, and one extra label applied to the padding tokens which should be masked for computing the loss.

are other evaluation metrics available for this type of task.

Different types of scenarios may be encountered:

1. The boundaries and the entity type (= label) match
2. The model found a not-existing entity
3. The model missed an entity
4. The model found the entity but assigned the wrong label
5. The model found the entity but got the boundaries wrong
6. The model found the entity but got both the boundaries and label wrong

Measures to find the above scenarios are calculated with nervaluate[1], a library that is typically used for extensive evaluation of Named Entity Recognition (NER) systems. Since NER is also a sequence tagging task, this can be applied similarly.

The following metrics are calculated:

- Message Understanding Conference (MUC) metrics:

  Correct: the entity is found and has the right label, it is a true positive

  Incorrect: the entity found and the ground truth entity do not match

  Partial: the entity found and the ground truth entity are somewhat similar

  Missed: a ground truth entity was missed by the model

  Spurious: the model found an entity that was not in the ground truth

- Possible: the number of ground truth annotations
  = COR + INC + PAR + MIS = TP + FN
- Actual: the number of annotations found by the tagger
  = COR + INC + PAR + SPU = TP + FP

[1] https://pypi.org/project/nervaluate

The calculations of the metrics can be more of less 'strict', both on the predicate boundaries found and on the label found:

- Type match: the tagged entity and the ground truth entity overlap partially and the label is the same
- Partial match: the found entity and ground truth entity overlap partially
- Strict match: the found entity and ground truth entity overlap exactly
- Exact match: the found entity and ground truth entity overlap exactly and also have the same label

5.3 LCM abstractness score

As mentioned in 2.4, computing a Linguistic Category Model score is a way to quantify the level of abstraction of a text. This can give a more general reflection of the text's level of abstraction than focusing on each of the sub-labels separately. It also provides a measure that can easily be used to compare the abstraction of differnt types of texts on a large scale. The labels are each given different weightings, where the lowest weight is assigned to the most concrete label and the highest weight to the most abstract label.

The LCM score as computed by Seih et al. (2017) is as follows:

$$LCMscore = \frac{DAV * 1 + IAV * 2 + SV * 3 + adjective * 4 + noun * 5}{DAV + IAV + SV + adjective + noun}$$

This formula is adapted as follows:

$$LCMscore = \frac{ACT * 1 + PRO * 2 + ATTAC * 3 + ATT * 3 + INNQU * 4}{ACT + PRO + ATTAC + ATT + INNQU}$$

where:

- ACT = Act
- PRO = Process
- ATTAC = Attitude + action
- ATT = Attitude
- INNQU = Innate quality

This score is meant to give an indication of the 'abstractness' of the text. Learned qualities are not included in the formula above because this category was not based on a category in the LCM and there is no reason to believe that (and if it does, where) it lies on the abstract-concrete spectrum. Nouns were left out because the nouns and adjectives are both captured as Quality. Attitude + action is judged to be similarly abstract as Attitude alone as they are both based on a State Verb.

It should be noted that this formula was originally used for word counts, where specific words were matched to a specific category. Because the approach was changed to detecting predicates, which are combinations of words, far less entities will be counted when using this formula. This could lead to instable measurements for short texts. It is therefore recommended to only process full job advertisements with this formula, not just a sentence or short paragraph.

6 Results and Analysis

Analysis can be done in three stages:

1. **Sequence classification:**

   Binary: finding all relevant predicates out of the correct predicates.

   Binary: find dispositional and behavioural predicates for all relevant and correct predicates.

   Multi-class: find all sub-labels for all relevant and correct predicates.

   (Sidenote: as said in the previous chapter, both count vectorized and TF-IDF vectorized features were tried. Because the scores from the models lie so close to each other for each task, only the results of the models using TF-IDF input vectors are given.)

2. **Three-step sequence tagging:** combine rule-based predicate segmentation with predicate classification.

3. **One-step sequence tagging:** find all predicates + their sub-label in-text through one network.

Stages 1 and 3 are applied to the annotated dataset and the results will be discussed in sections 6.1 and 6.3. For these stages, the outcomes of the different algorithms are elaborated on and illustrated with an example sentence or predicate from the test set.

Stage 2 is not applied and will not be evaluated in this chapter, because the annotated dataset is not suitable for this task. The steps involved in this approach are shortly revised in section 6.2. It will be applied in chapter 7.

6.1 Sequence classification

6.1.1 Binary classification: relevant vs. not relevant

Before classifying all predicates into the (sub)classes of dispositional and behavioural, it is important to filter out those predicates that are not relevant for this classification task. The results given in this section show to what extent it is possible to automate this.

6.1.1.1 Model performance

Notation: performance on validation set — performance on test set

	DT	**NB**	**SVM**	**RF**	**XGB**
Accuracy	.71 — .74	.76 — .79	.75 — .79	.74 — .79	.74 — .78
Precision	.71 — .75	.76 — .79	.75 — .79	.74 — .79	.74 — .78
Recall	.71 — .74	.76 — .79	.75 — .79	.74 — .79	.74 — .78
F1	.69 — .73	.76 — .79	.74 — .78	.74 — .78	.74 — .77
Mean AUROC	.76 — .76	.85 — .87	.84 — .86	.84 — .86	.83 — .84
Mean AUPRC	.78 — .78	.85 — .87	.83 — .86	.83 — .86	.82 — .83

	Word2Vec	**BERT**	**RoBERTa**
Accuracy	.82 — .81	.84 — .84	.86 — .88
Precision	.82 — .81	.84 — .84	.86 — .88
Recall	.82 — .81	.84 — .84	.86 — .88
F1	.82 — .81	.84 — .84	.86 — .88
Mean AUROC	.89 — .89	.93 — .92	.93 — .94
Mean AUPRC	.89 — .89	.93 — .91	.94 — .94

Table 10: Performance evaluation scores of all classifiers (weighted averages).

RoBERTa (RobBERT) outperforms the other models in every aspect.

6.1.1.2 ROC/PR curves

The ROC curve in figure 17 shows how well the relevant samples can be distinguished from the non-relevant samples by each model. RoBERTa shows the best performance for this task with an AUROC of .94.

The PR curves in figure 18 show for the two classes separately how tough they are to distinguish by the model. The scores for Not relevant are lower, as is the baseline because there are less samples available for this class than for Relevant.

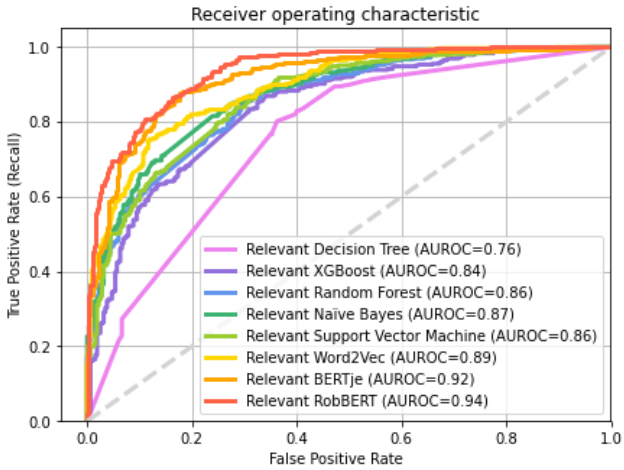


Figure 17: Receiver Operator Characteristic curve found for the Relevant set of predicates.

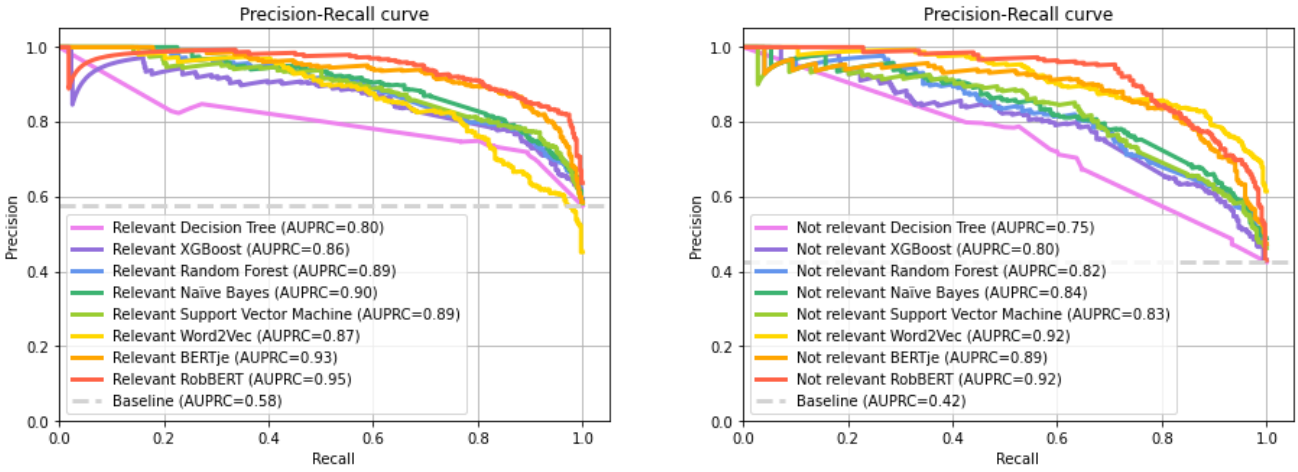


Figure 18: Precision-recall curves found for the relevancy classifier. On the left: Relevant. On the right: Not relevant.

6.1.1.3 Feature importance (as given by XGBoost)

The feature importance was calculated by computing the mean of the TF-IDF scores of every word, per class. A high TF-IDF score means that the feature (word) carries more significance for the class. That is, it is a common feature on document-level (predicate) but rare on collection-of-documents level. Looking at the feature importance can give hints on why some classes have a higher classification performance than others. The given feature importance tables in this chapter were found by the XGBoost classifier, and they are expected to be quite stable across all classifiers that used TF-IDF input vectors as they all make use of the same corpus.

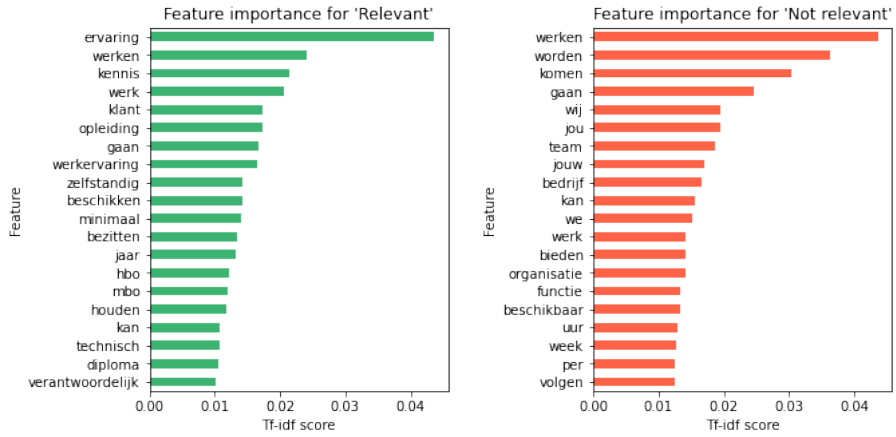


Figure 19: The most important features in predicates labeled as Relevant or Not relevant as found by XGBoost. The y-axis gives the feature and the x-axis gives the average TF-IDF value for each feature for all predicates with the corresponding label.

The most important features found for the relevant predicates show a combination of skills ("kennis", "ervaring", "hbo", "diploma"). The most important features for the not-relevant predicates show words that can be associated with the company ("team", "bedrijf", "organisatie"), working hours ("uur", "week", "beschikbaar"). Those words are indeed expected to be filtered out by this classification task.

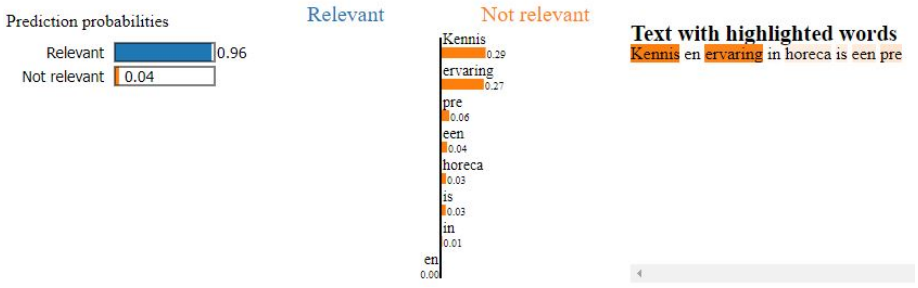
6.1.1.4 Example predicates

Example sentence in "Relevant

Predicate text: "Kennis en ervaring in horeca is een pre"
TF-IDF features: ['ervaring' 'kennis' 'pre']
TF-IDF weights: [0.49185614 0.56302231 0.66414111]

Support Vector Machine

BERT



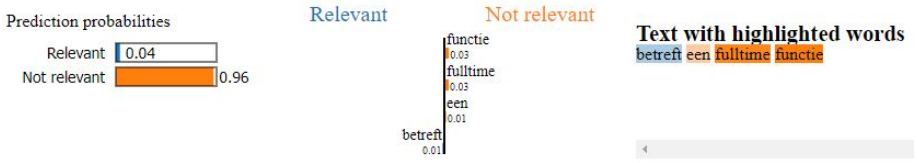
This predicate seems to be easy for all models. LIME shows that all models associate "ervaring" and "kennis" with the relevant class, and indeed both of these words were found as important features for the relevant class.

Example sentence in "Not relevant"

Predicate text: "betreft een fulltime functie"
TF-IDF features: ['betreffen' 'fulltime' 'functie']
TF-IDF weights: [0.65014438 0.53726729 0.53726729]

Support Vector Machine

BERT



All models again predicted Not relevant, due to the importance of "fulltime" for this decision. The SVM and BERT both show a slight hesitation due to the word "betreffen" which is a verb, which might suggest the sentence being relevant. But its influence is only small.

6.1.2 Binary classification: dispositional vs. behavioural

After finding all relevant predicates, the next binary classification task is to classify the relevant predicates into the (sub)classes of dispositional and behavioural. These results show to what extent it is possible to automate this with a binary classifier.

6.1.2.1 Model performance

Notation: performance on validation set — performance on test set

	DT	**NB**	**SVM**	**RF**	**XGB**
Accuracy	.83 — .82	.83 — .85	.86 — .86	.85 — .84	.84 — .81
Precision	.84 — .83	.83 — .85	.86 — .86	.85 — .85	.84 — .82
Recall	.83 — .82	.83 — .85	.86 — .86	.85 — .84	.84 — .81
F1	.83 — .82	.83 — .85	.86 — .86	.85 — .84	.84 — .81
Mean AUROC	.85 — .86	.93 — .92	.93 — .92	.93 — .91	.91 — .88
Mean AUPRC	.87 — .86	.91 — .91	.91 — .91	.92 — .90	.89 — .86

	Word2Vec	**BERT**	**RoBERTa**
Accuracy	.87 — .86	.90 — .90	.89 — .88
Precision	.88 — .86	.90 — .90	.90 — .90
Recall	.87 — .86	.90 — .90	.89 — .88
F1	.88 — .86	.90 — .90	.89 — .89
Mean AUROC	.94 — .93	.96 — .96	.95 — .94
Mean AUPRC	.93 — .92	.95 — .95	.94 — .92

Table 11: Performance evaluation scores of all classifiers (weighted averages).

Table 11 shows that, in general, all models are capable of distinguishing between dispositional and behavioural predicates. BERT outperforms every other model, although RoBERTa is very close.

6.1.2.2 ROC/PR curves

The ROC curve in figure 20 shows how well the dispositional ("Being") samples can be distinguished from the behavioural ("Doing") samples by each model. BERT shows the best performance for this task with a AUROC of .96. This means that a sample classified as behavioural has a probability of .96 of actually being so.

The PR curves in figure 21 show for the two classes separately that the dispositional ("being") phrases are easier to detect by the model than those that are behavioural. They generally have a higher precision and recall scores than the behavioural class. The number of samples is also higher, though.

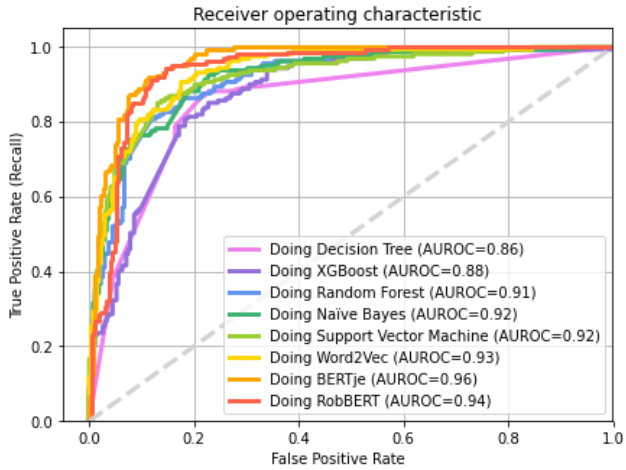


Figure 20: Receiver Operator Characteristic curve found for the set of behavioural predicates.

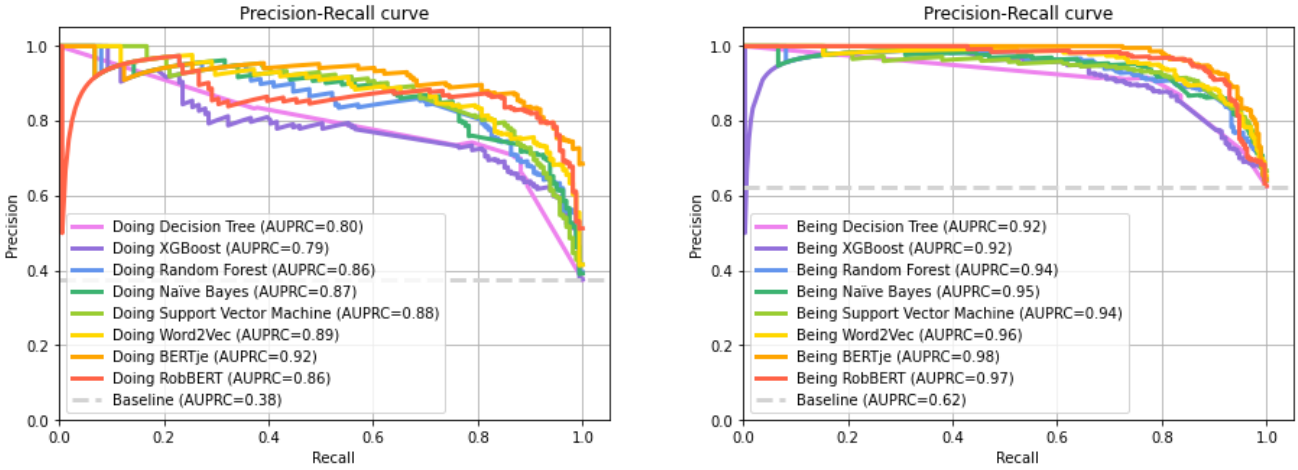


Figure 21: Precision-recall curves found for the classification of dispositional and behavioural predicates. On the left: behavioural. On the right: dispositional.

6.1.2.3 Feature importance

As shown in figure 22, the most important features of the class “Doing” contain many words that are quite general but directly work-related (“uitvoeren”, “werkzaamheid”, “verwerken”, “project”, “werken”). This was expected. Job-specific words are possibly too uncommon in the feature set to be seen as important, or too rare to be in the feature set at all. The ”Being” class contains many important features that are words related to common learned skillsets (“ervaring”, “kennis”, “werkervaring”, “opleiding”, “hbo”, “mbo”, “diploma”).

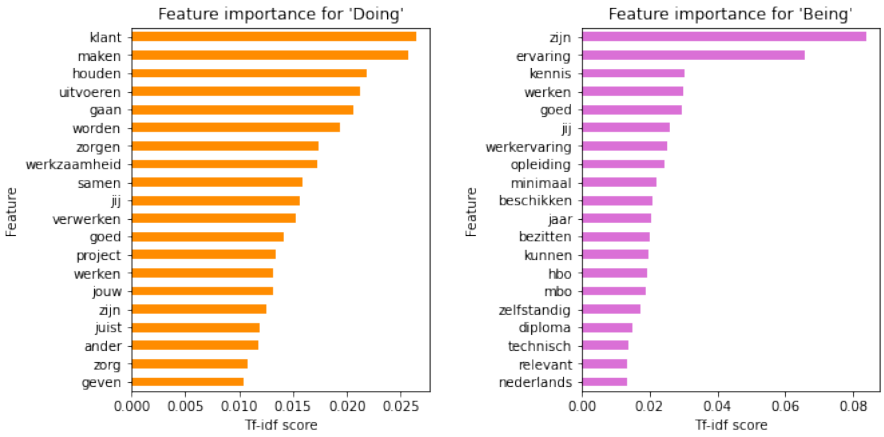


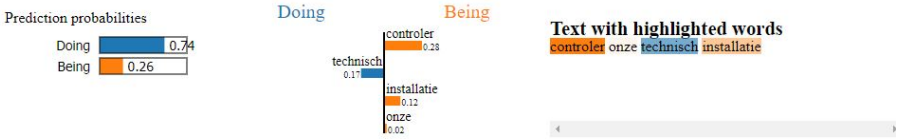
Figure 22: The most important features in dispositional and behavioural predicates as found by XGBoost. The y-axis gives the feature and the x-axis gives the average TF-IDF value for each feature for all predicates with the corresponding label.

6.1.2.4 Example predicates

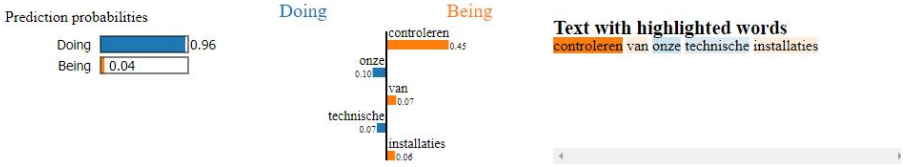
Example sentence in "Behavioural"

Predicate text: "controleren van onze technische installaties"
TF-IDF features: ['controler' 'installatie' 'onze' 'technisch']
TF-IDF weights: [0.51856845 0.51856845 0.51856845 0.43961381]

Naïve Bayes



Word2Vec



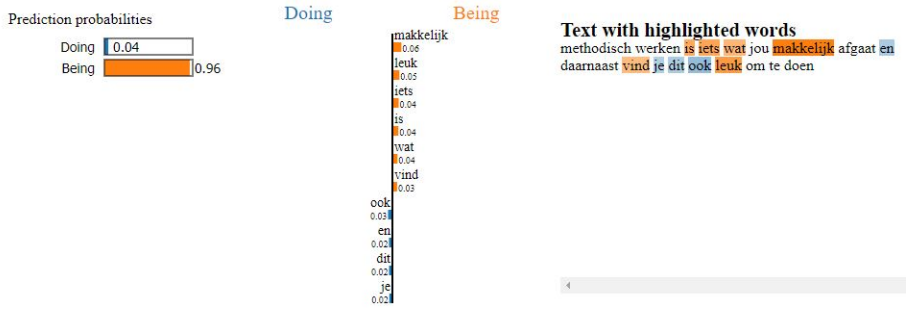
All models were correct for this example. This is a good example where both a word that typically suggests a dispositional phrase occurs together with a word that typically suggests a behavioural predicate. "Technische" often is written in the sense of "technische opleiding" "technisch inzicht". In this case, "controleren" was rightly found as carrying more weight. It is the main verb of the predicate.

Example sentence in "Dispositional"

Predicate text: "Methodisch werken is iets wat jou makkelijk afgaat en daarnaast vind je dit ook leuk om te doen"
TF-IDF features: ['doen' 'leuk' 'makkelijk' 'vinden' 'werken']
TF-IDF weights: [0.49878559 0.45087115 0.49878559 0.41218719 0.35950355]

Naïve Bayes

Word2Vec



All models were correct for this example. All give most weight to the words "vinden", "leuk", and "makkelijk". This makes sense because these words are nearly always about a person, and their attitude towards an action.

6.1.3 Multi-class classification: all subcategories

Another way to classify the relevant predicates is to do multi-class classification of the LCM sub-labels. Because this task is more complex than binary classification, the performance is expected to be lower than for the previous task.

6.1.3.1 Model performance

Notation: performance on validation set — performance on test set

	DT	**NB**	**SVM**	**RF**	**XGB**
Accuracy	.60 — .58	.61 — .63	.64 — .65	.63 — .63	.59 — .57
Precision	.62 — .58	.59 — .61	.65 — .65	.64 — .62	.65 — .64
Recall	.60 — .58	.61 — .63	.64 — .65	.63 — .63	.59 — .57
F1	.60 — .57	.58 — .59	.64 — .64	.61 — .61	.60 — .58
Mean AUROC	.82 — .79	.90 — .90	.89 — .90	.89 — .89	.87 — .86
Mean AUPRC	.60 — .56	.68 — .66	.66 — .68	.67 — .66	.61 — .60

	Word2Vec	**BERT**	**RoBERTa**
Accuracy	.67 — .65	.71 — .69	.69 — .69
Precision	.67 — .64	.70 — .68	.69 — .70
Recall	.67 — .65	.71 — .69	.69 — .69
F1	.66 — .64	.70 — .68	.69 — .70
Mean AUROC	.92 — .91	.92 — .92	.90 — .91
Mean AUPRC	.71 — .69	.73 — .72	.70 — .71

Table 12: Performance evaluation scores of all classifiers (weighted averages).

Table 12 shows that, as expected, the models that used pre-trained word embeddings, are better at this classification task. Especially BERT, which was fine-tuned to the task, shows the best scores.

6.1.3.2 ROC/PR curves

Looking at the sub-labels separately, for BERT, we get the ROC and PR curves in figure 23. The ROC and PR curves in multi-class problems give one curve for each label, where the task is treated as a binary 'one label versus the rest' problem. This makes the graph slightly more tricky to interpret than if it already were a binary problem. The initial threshold for assigning a positive label to a sample is .5. However, this threshold can be varied, to be more or less 'strict' in this decision. This has been done for both finding the ROC and PR curves as seen above. The Area Under the Curve then gives the probability of being correct when assigning a positive label to a sample, at every threshold. There are slight differences in AUROC seen in the ROC curves. The ROC graph shows that the classifier performs better than random classification for all labels (between 84% and 98% of being correct), when treated as a binary problem.

However, the difference in classification performance between classes become more evident in the PR curves. The graph shows a trend that is seen in every model that was tried. The Attitude (+ action) classes show the lowest AUPRC scores, while the qualities show the highest AUPRC scores. This should be interpreted as follows. For each

label, it gives the trade-off between precision and recall with a certain threshold for determining if a sample belongs to this specific class or not. If we aim for a high precision, this threshold needs to be higher, which means that less samples will be found to be positive. This gives a lower recall. Vice versa, aiming for a higher recall means lowering the threshold, which will give lower precision scores. The smaller the effect of this trade-off is, the more this curve approaches (1,1) in the graph, the higher the AUPRC score will be, and the better the classifier is at distinguishing this class from the rest of the classes.

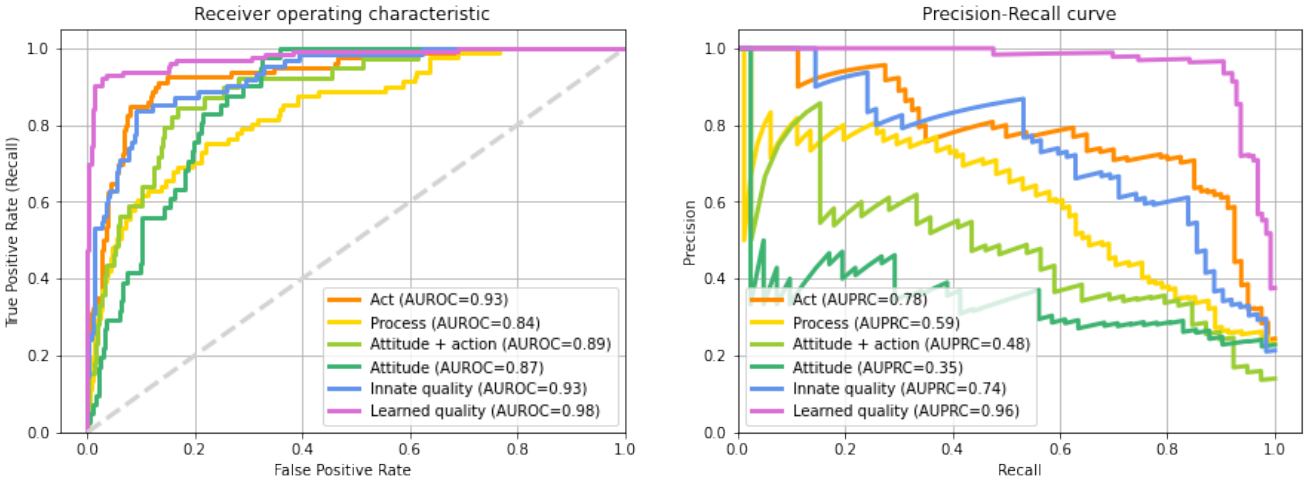


Figure 23: On the left: Receiver Operator Characteristic curves (one-versus-rest) for all relevant sub-labels. On the right: Precision-recall curves (one-versus-rest) for all sub-labels. The classifier was BERT.

With this reasoning, it can be stated that the class of Learned qualities is the easiest class for the classifier to distinguish. The Attitude and Attitude + action classes are the hardest classes to find.

6.1.3.3 Feature importance

The Learned qualities had the highest AUPRC score. The highest given features, given in figure 24, are words that clearly relate to education or work experience. This seems to be a set of words that are, indeed, not likely to occur in predicates that are *not* about a Learned quality and are thus not easily confused for another class. The features for Innate qualities show that "zijn" is an important word that occurs relatively frequently in this class. This is expected, since Innate qualities belong to the dispositional side, which means that the predicate is about what a person *is*. Someone *is* ["flexibel"/"nauwkeurig"/"zelfstandig"/etc.]. Also less verifiable skills belong to this class, for example "Je kan goed luisteren", which explains why "kan" and "kunnen" have a relatively high score.

The most important features for Act and Process are given in figure 25. These classes are expected to be rather specific to the type of job, as they include the activities that are directly related to the work. One theory for why these classes are harder to classify than the Qualities is that Qualities are more generalizable. Skills and personal characteristics are relevant to mention in every kind of job, whereas concrete actions are more

job-specific. This gives more variability to how this class is expressed. Another theory is that Processes and Acts can be easily confused. They both contain the words “gaan”, “maken”, “klant”, “worden”, “zorgen”, and “werkzaamheid”. The difference between an Act and Process might be more open for interpretation. This was already seen when comparing the classifications made by different annotators.

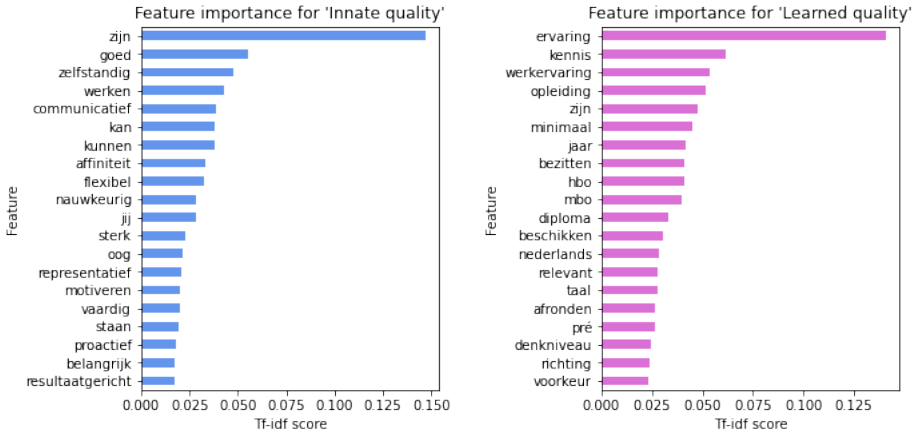


Figure 24: The most important features in predicates labeled as Innate quality or Learned quality as found by XGBoost. The y-axis gives the feature and the x-axis gives the average TF-IDF value for each feature for all predicates with the corresponding label.

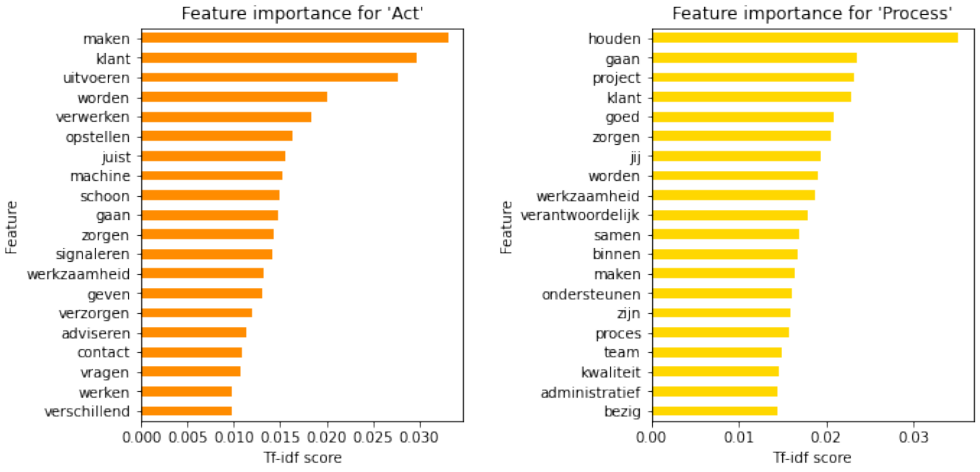


Figure 25: The most important features in predicates labeled as Act or Process as found by XGBoost. The y-axis gives the feature and the x-axis gives the average TF-IDF value for each feature for all predicates with the corresponding label.

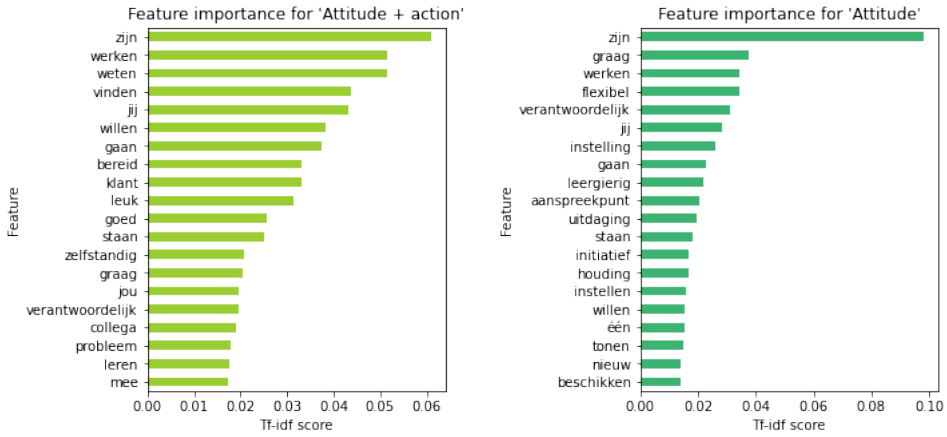


Figure 26: The most important features in predicates labeled as Attitude (+ action) as found by XGBoost. The y-axis gives the feature and the x-axis gives the average TF-IDF value for each feature for all predicates with the corresponding label.

The Attitude and Attitude + action classes are the hardest ones to detect for all classifiers. This is a logical observation, given that these are in the middle of the concrete-abstract spectrum, so they may contain information about an action as well as about a feeling towards an action expressed by "zijn", which was also an important feature for Innate quality. Because of the overlap, not the features alone but the combination of features are expected to clarify what class the samples belongs to.

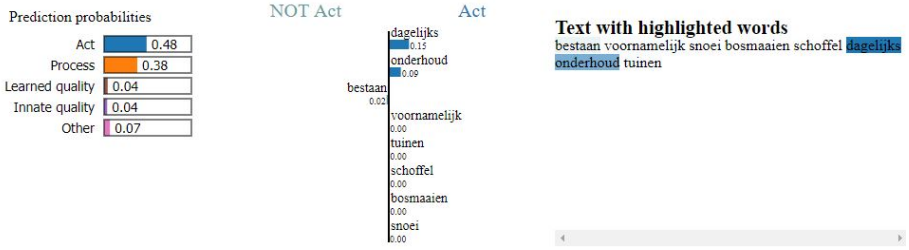
6.1.3.4 Example predicates

Example sentence in "Act"

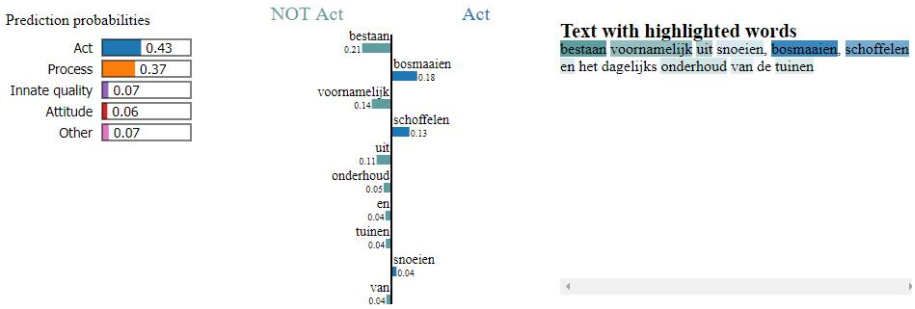
Predicate text: "bestaan voornamelijk uit snoeien, bosmaaien, schoffelen en het dagelijks onderhoud van de tuinen"
TF-IDF features: ['bestaan' 'dagelijks' 'onderhoud']
TF-IDF weights: [0.57735027 0.57735027 0.57735027]

All models show doubt between Act and Process for this example. It seems that "bestaan" suggests a Process (at the "NOT act" side for RoBERTa). It is notable that in the examples above only RoBERTa finds the words "bosmaaien", "schoffelen", and "snoeien". This shows the advantages of using a pre-trained model, or at least of a model that has more vocabulary than the words in the training set. The verbs are quite specific to this type of job, they capture the Act in this predicate.

Naïve Bayes



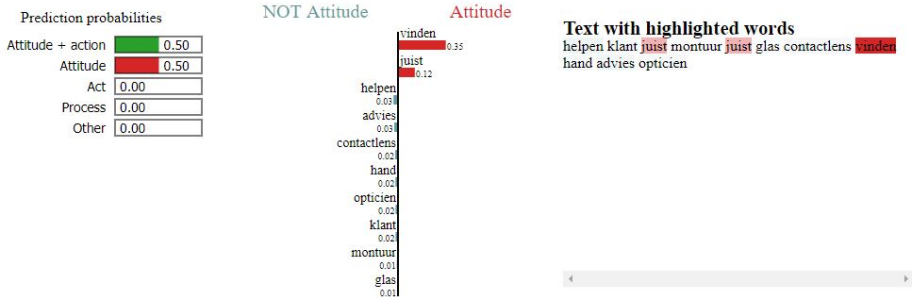
RoBERTa



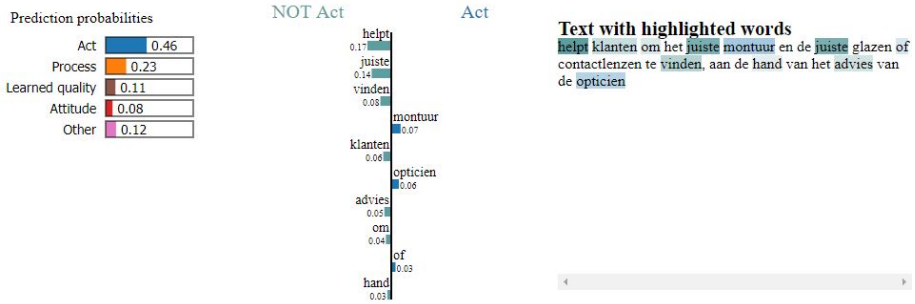
Example sentence in "Process"

Predicate text: "helpt klanten om het juiste montuur en de juiste glazen of contactlenzen te vinden, aan de hand van het advies van de opticien"
TF-IDF features: ['hand' 'helpen' 'juist' 'klant' 'vinden']
TF-IDF weights: [0.34716673 0.36783649 0.73567297 0.3325013 0.30397327]

Decision Tree



RoBERTa



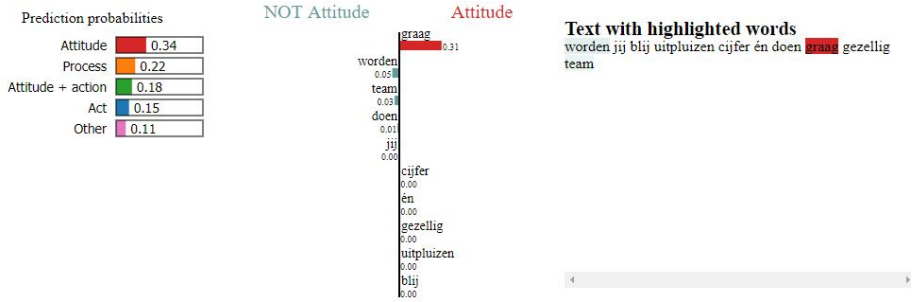
This example proves to be difficult for the models. The TF-IDF words do suggest labels on the behavioural half, but there is variation in the chosen labels. The Decision Tree seems to have found a path along several relevant words, however does not end up in the right leaf node. RoBERTa finds the strongest associations to be between "montuur" and "opticien" and the category Act. Indeed, if the predicate would have been "kiest

een montuur bij de opticien", then it could have been an act. However, "helpen" should be noticed as most important verb in this predicate.

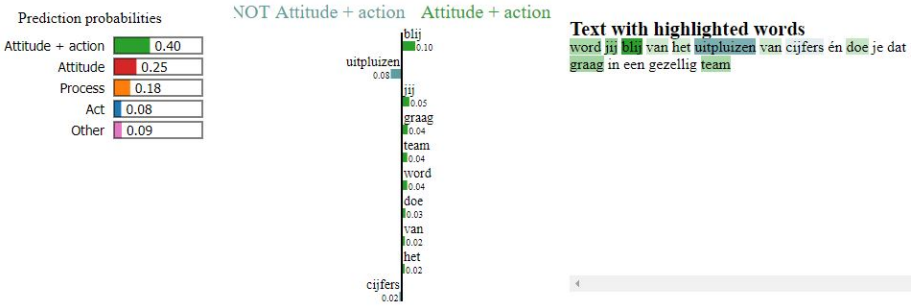
Example sentence in "Attitude + action"

Predicate text: "word jij blij van het uitpluizen van cijfers én doe je dat graag in een gezellig team"
TF-IDF features: ['doen' 'gezellig' 'graag' 'jij' 'team' 'worden']
TF-IDF weights: [0.44113665 0.44113665 0.36454716 0.33159483 0.41634795 0.44113665]

Random Forest



BERT



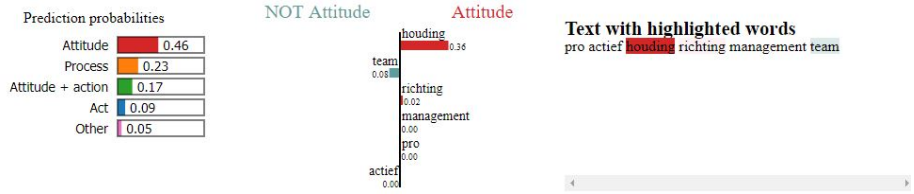
In this example, only BERT and RoBERTa strongly associate the word "blij" with an Attitude + action label. This word is missing in the TF-IDF vector, which makes that the Random Forest lacks this knowledge.

Example sentence in "Attitude"

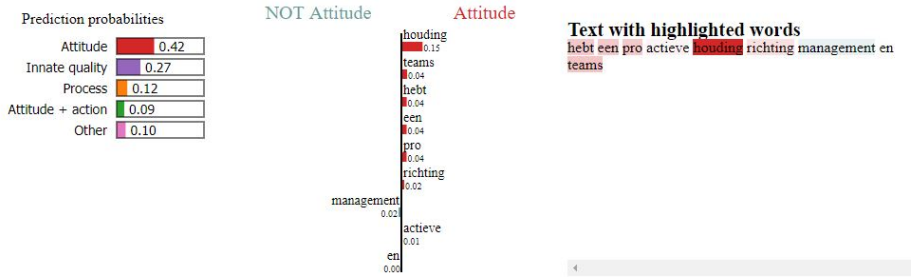
Predicate text: "hebt een pro actieve houding richting management en teams"
TF-IDF features: ['actief' 'houding' 'pro' 'richting' 'team']
TF-IDF weights: [0.453828 0.4808482 0.453828 0.38846552 0.453828]

All of the models find the strong association between "houding" and the Attitude category.

Support Vector Machine



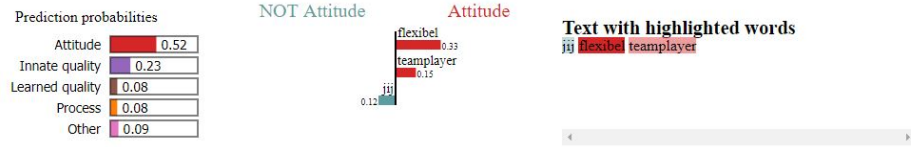
Word2Vec



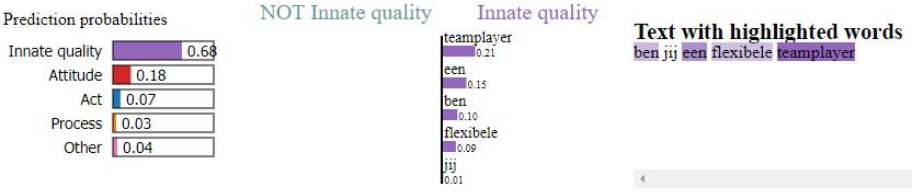
Example sentence in "Innate quality"

Predicate text: "ben jij een flexibele teamplayer "
TF-IDF features: ['flexibel' 'jij' 'teamplayer']
TF-IDF weights: [0.54249291 0.50475914 0.67150551]

XGBoost



RoBERTa



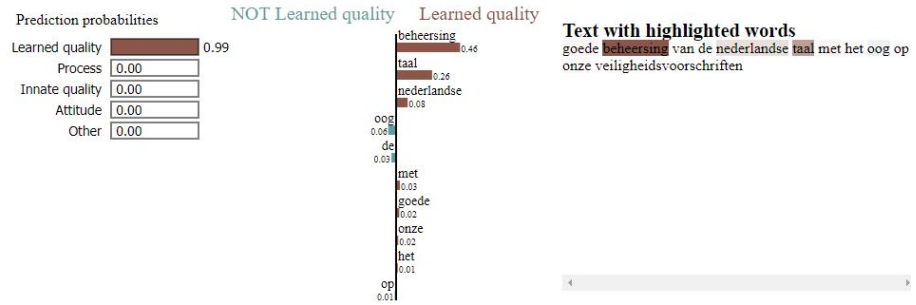
All models except XGBoost predicted this predicate correctly. Notable is that the word "ben" is not correctly lemmatized because "zijn" is a word that is known by the TF-IDF vectorizer, and was found as one of the most important features, as also shown by RoBERTa.

Example sentence in "Learned quality"

Predicate text: "Goede beheersing van de Nederlandse taal(met het oog op onze veiligheidsvoorschriften)"
TF-IDF features: ['beheersing' 'goed' 'nederlands' 'onze' 'oog' 'taal']
TF-IDF weights: [0.38229335 0.33738478 0.36013925 0.47317578 0.49404585 0.37728822]

Naïve Bayes

Word2Vec



This example was rather easy to identify for all models. It has many strongly indicative words for a Learned quality, like "Nederlands", "beheersen", and "taal".

6.1.3.5 Error analysis

Even though we are doing multi-class classification, one observation is missing from the evaluation so far. The classes are not entirely independent of each other, they have an ordering from least abstract to most abstract (except for the Learned qualities, which are dispositional like Innate qualities but not necessarily abstract). Therefore, it is important to investigate the errors that are made. This can be seen in the confusion matrices in figure 27.

The confusion matrices show that the Random Forest classifier gives most incorrect predictions for the Act class. BERT does show confusion between the Act and Process labels, however less so between the extremes. The diagonal shows a gradual decrease of errors towards classes that are further away, which means that the BERT classifier mostly confuses true classes for one of their neighbouring classes. Therefore, the incorrect predictions that the BERT classifier gives are still closer to their true class than the errors that the Random Forest classifier gives.

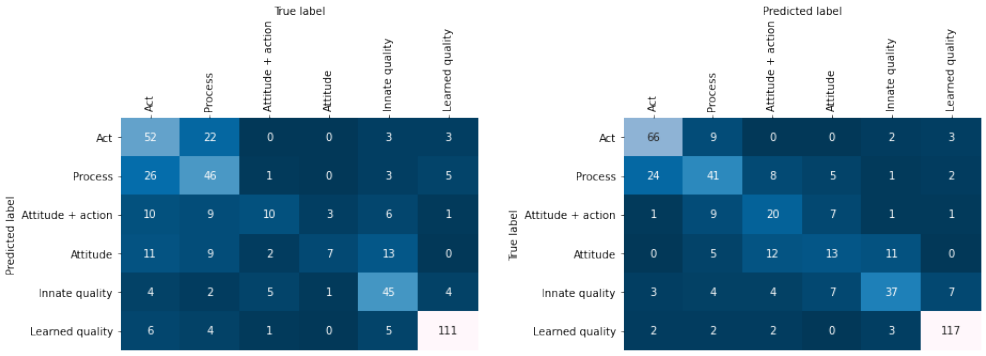


Figure 27: Confusion tables for two classifiers for all relevant sub-labels. One the left: Random Forest. On the right: BERT.

6.1.3.6 Multi-class to binary

It would also be interesting to see if training the classifier by the sub-labels would help in determining whether the given predicates are dispositional or behavioural. Therefore the multi-class test labels were converted back to dispositional (merging Attitude and Quality) and behavioural (merging Act and Process) labels and compared to the results of the previous task as discussed in section 6.1.2. Both models are BERT.

Binary classification				**Multi-class to binary classification**				
	Precision	**Recall**	**F1**		**Precision**	**Recall**	**F1**	**support**
Behavioural	.85	.89	.87	**Behavioural**	.84	.87	.86	161
Dispositional	.93	.90	.92	**Dispositional**	.92	.90	.91	268
Accuracy			**.90**	**Accuracy**			**.89**	429
Macro avg	.89	.90	.89	**Macro avg**	.88	.89	.88	429
Weighted avg	.90	.90	.90	**Weighted avg**	.89	.89	.89	429

Table 13: Performance evaluation scores of two classifiers. On the left: the binary classifier for labels Behavioural and Dispositional. On the right: the multi-class classifier after merging the sub-labels into the labels Behavioural and Dispositional.

The results are seen in table 13. There is only a very slight difference seen: the accuracy of the merged multi-class classifier is .89 while it was .90 for the binary classifier. This difference is not significant, therefore, the sub-labels did not help to separate this binary split. Converting from multi-class to binary will give similarly confident results as the model trained for binary classification.

6.2 Three-step sequence tagging

For three-step sequence tagging, the approach is as follows:

1. Determine the boundaries of the predicates within a sentence
2. Classify each of the found predicates as relevant or not relevant
3. Classify each of the relevant predicates as member of one of the subclasses

This token tagging approach can not be evaluated fairly with the annotated dataset at hand. This is because the predicate boundaries were already found by using the same method for this data. When applied, the predicate boundaries would match exactly and that would bring this task back to a classification task. The analysis of the classifiers can already be found in section 6.1. This approach will be applied to two new example texts in chapter 7.

6.3 One-step sequence tagging

For in-text labeling, the Dutch pre-trained BERT model 'BERTje' was fine-tuned to a sequence tagging task.

6.3.1 Token-level evaluation

Because BERT makes use of the WordPiece tokenization, it also classified the text with a label for every token. These classifications were adapted to word level by applying the label for the last WordPiece to the whole word. On word level, an accuracy of .75 was obtained, as can be read in table 14.

	Precision	**Recall**	**F1**	**Support**
O	.88	.89	.89	5788
B-Act	.51	.29	.37	80
I-Act	.47	.41	.44	499
B-Process	.42	.27	.33	81
I-Process	.48	.48	.48	706
B-Attitude + action	.36	.13	.19	39
I-Attitude + action	.36	.35	.35	385
B-Attitude	.29	.15	.19	41
I-Attitude	.28	.21	.24	262
B-Innate quality	.52	.53	.53	62
I-Innate quality	.44	.70	.54	275
B-Learned quality	.71	.50	.59	126
I-Learned quality	.75	.77	.76	944
Accuracy			**.75**	9288
Macro avg	.50	.44	.45	9288
Weighted avg	**.75**	**.75**	**.75**	9288

Table 14: Evaluation of token-level sequence tagging performance.

In the end, we are not interested in the "I-" and "B-" prefixes. Therefore, these classes were merged. This is a simplification step. Therefore, this slightly improved the accuracy from .75 to .76. This can be read in table 15

	Precision	**Recall**	**F1**	**Support**
O	.88	.89	.89	5788
Act	.52	.44	.48	579
Process	.49	.47	.48	787
Attitude + action	.37	**.33**	.35	424
Attitude	.29	**.20**	.24	303
Innate quality	.49	.72	.58	337
Learned quality	**.77**	**.77**	.77	1070
Accuracy			**.76**	9288
Macro avg	.54	.55	.54	9288
Weighted avg	**.75**	**.76**	**.76**	9288

Table 15: Evaluation of token-level sequence tagging performance with the 'BIO' prefixes removed.

The baseline accuracy for this task was .63 based on the most frequent class, so the model performs better than the baseline.

Figure 28 shows the confusion in word-level tagging of the test set:

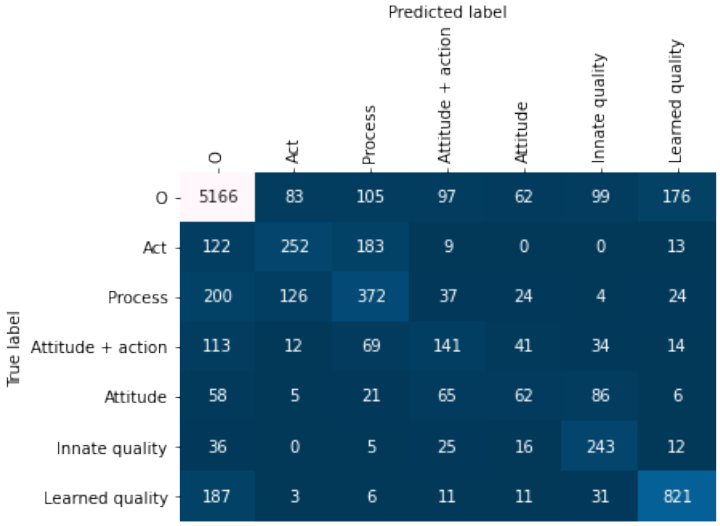


Figure 28: Confusion table for word-level sequence tagging of the test set

As can be seen in both the confusion table in figure 28 and in the performance scores for every label, the "O" class gives the highest scores. This is the text that was not tagged. This constitutes of all text that is not relevant, which is the biggest class since not all sentences contain relevant predicates. The confusion table shows most confusion between the words tagged "O" and words not tagged as "O", suggesting either missed predicates (FN), incorrectly tagged predicates (FP) or errors in the boundaries of the tagged predicates.

6.3.2 Entity-level evaluation

Besides the given scores on word level, it would be desirable to scale this to entity-level and find out about not only the class prediction per word but also the accuracy of the entity boundary predictions.

	Type	**Partial**	**Strict**	**Exact**
Correct	**255**	169	134	169
Incorrect	122	0	243	208
Partial	0	208	0	0
Missed	54	54	54	54
Spurious	469	469	469	469
Possible	431	431	431	431
Actual	**846**	**846**	**846**	**846**
Precision	.30	.32	.16	.20
Recall	**.59**	.63	**.31**	.39
F1	.40	.43	.21	.26

Table 16: Nervaluate results for all 846 predicates

The results in table 16 are to be interpreted in the following way.

Looking at the table, out of the 846 predicted predicates, 255 were correct in label and partially correct in entity boundaries. The focus of this study, however, is on recall, because we want to retrieve as many of the annotated entities as possible. Thus, we can conclude that 59% of the predicates were retrieved, if our requirements are that there is at least partial overlap between the true and predicted label and that the predicted predicate label is correct. If it is required to have an exact overlap, this score reduces to 31%.

There is a large variance between the performance of the model for different types of predicates. From the classification table and the scores table, it is notable that out of the relevant labels, the Learned quality predicates are best recognized by the model, with a precision and recall score of .77. The Attitude and Attitude + action predicates have the worst classification scores, namely a recall of .20 and .33 respectively. This is in line with the scores of the classification task. Both classes are discussed below.

Table for Learned qualities:

	Type	**Partial**	**Strict**	**Exact**
Correct	107	62	61	62
Incorrect	11	0	57	56
Partial	0	56	0	0
Missed	8	8	8	8
Spurious	64	64	64	64
Possible	126	126	126	126
Actual	182	182	182	182
Precision	**.59**	.49	.34	.34
Recall	**.85**	.71	**.48**	.49
F1	.69	.58	.40	.40

Table 17: Nervaluate results for all 182 predicates labeled as Learned quality

Table for Attitude:

	Type	**Partial**	**Strict**	**Exact**
Correct	9	12	2	12
Incorrect	28	0	35	25
Partial	0	25	0	0
Missed	5	5	5	5
Spurious	89	89	89	89
Possible	42	42	42	42
Actual	126	126	126	126
Precision	**.07**	.19	.02	.10
Recall	**.21**	**.58**	**.05**	.29
F1	.11	.29	.02	.14

Table 18: Nervaluate results for all 126 predicates labeled as Attitude

Table 17 shows that 85% of the predicates annotated as Learned quality were found, and about half of the found Learned qualities was also an exact match in boundaries.

The precision of entity type is .59, which means that 59% of the predicted Learned qualities was partially correct, and about half of those were also strictly correct.

At the other end is the label Attitude, which only reaches a precision and recall respectively up to .29 and .20. From table 18 it becomes clear that 58% of the actual Attitude predicates were partially found, however they were not given the right entity type label. Only 5% of the true Attitude predicates were both an exact match in boundaries as well as given the right label. When it comes to precision, only 7% of the predicted Attitude predicates that had partial overlap with a ground truth predicate, also had the right label. That means that, even though 21% of the Attitude predicates were found, only 7% of the predicted Attitude predicates were partially correct. This suggests that the model predicted many more Attitude predicates than there actually are in the data. The confusion matrix confirms this suggestion.

This analysis shows a big difference in performance between the different labels, similar to what was found in the predicate classification tasks.

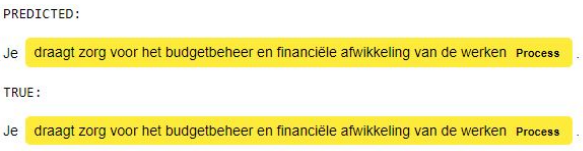
6.3.3 Examples

In this section, some example predictions from the validation and test set are shown to investigate the variety of errors that can occur in a sequence tagging task.

The following are examples of texts that were given the right labels:

The tagging below is correct, it is an example of an exact match.

The tagging below is correct, it is another example of an exact match.



The tagging below is correct, it is another example of an exact match. This is not entirely true. The text in English describes a Learned quality. Because the tagger was trained on Dutch text only, and the annotations also do not capture English predicates, this was not found.

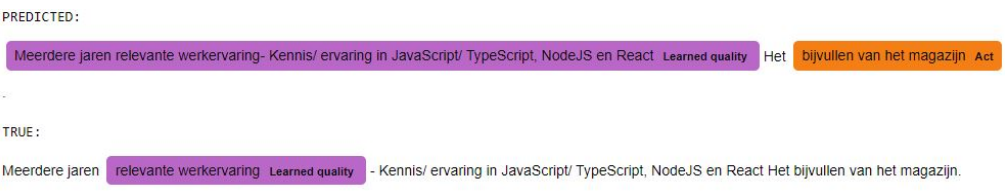
The example below shows a type match. The label is correct and there is overlap, but the boundaries are not the same.

The tagging below was correct, this text is not relevant. *Houden van* English: 'to love' often introduces an Attitude. However, this text is about a company so it should not be labeled. This would have been a false positive otherwise, but now it is a true negative.

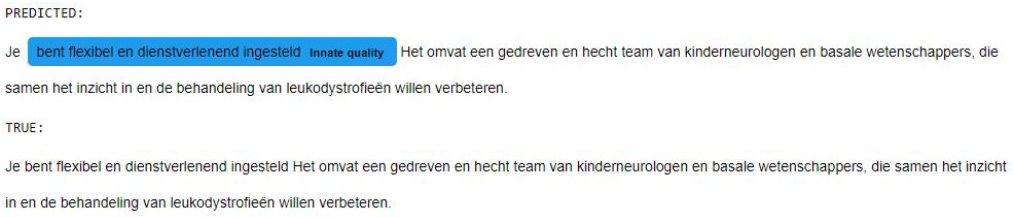
The text below shows three tagged acts for one annotated act. In fact, the predicted boundaries are better. The first one is seen as a spurious tag, although it is actually correct. The second and third are seen as type matches in evaluation.

In the following texts, the predicted label was different than the 'true' label. These examples are interesting to look at.

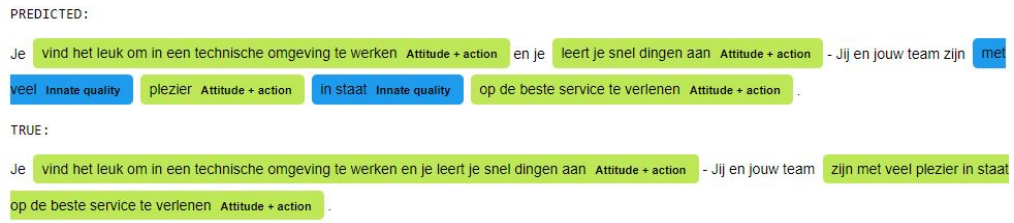
In the example below, the predicted tagging is better than the annotated tags. The Learned quality captures the entire skill, and the 'spurious' Act tag should have been in the annotated set.



The text below shows a tag that is seen as spurious, however it is a correct prediction of an Innate quality.



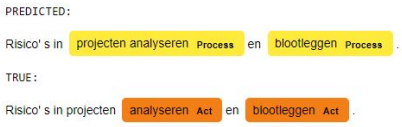
The example below shows that the tagger sometimes tends to make smaller chunks than it should.



The example below is interesting. It actually is quite right, tagging both an Attitude and a Process, which adds up to Attitude + action. It got it correct, but did not attach the right label.

The example below shows a missed tag.

To begin with, the text below is not grammatical. Ungrammatical text is hard to deal with, especially when detecting boundaries of the predicates. In fact, this tag should have started with, for example, *Je zal* English: 'you will'. In that case, the predicted predicate should have been *zal risico's analyseren* English: 'will analyze risks'. This was also not correct in the annotated text. Far more profound grammar rules should account for this type of structure. Secondly, the given labels are not right. These tags count as partial matches. However, whether they really are an Act or Process is questionable. One might considered this to be an edge case.



The example below shows a prediction that fits better than the annotated label. *Verantwoordelijk zijn* English: 'being responsible' is an Attitude, where *ontzorgen* English: 'unburden' would be the related action. The boundaries, however, were better in the annotated labeling.

Examples like those given above show that even the annotations that were provided are absolutely not foolproof. Still, they are meant to guide the model to obtaining the right labels, and prove to be useful even if they are not exactly a right fit. It should also be noted that these are examples that stood out. The examples given above are not necessarily representative for the entire set of predicted texts. Both the predictions and the tags as annotated show errors.

7 The detectors in practice: tagging example texts

Having tried both sequence classification and sequence tagging, this chapter will compare the two proposed tagging methods as applied two full job advertisement texts, by applying the one-step and three-step approaches as defined in chapter 6. So far we have only looked at token, word, predicate, and sentence-level results but this chapter in intended to indicate what approach is best used in practice. For consistency, the models trained on BERT were used for both the classifiers and the tagger.

To find the best ground truth tagging for the two texts, they were also labeled manually by two annotators. The fully labeled texts are found in Appendix **C**.

7.1 Evaluation

The following results were obtained for word-level tagging of the two approaches:

One-step sequence tagging

	Precision	Recall	F1
O	.84	.92	.88
Act	.62	.64	.63
Process	.27	.38	.31
Attitude + action	.75	.34	.47
Attitude	.40	.38	.39
Innate quality	.65	.51	.57
Learned quality	.82	.84	.83
Accuracy			**.77**
Macro avg	.62	.57	.58
Weighted avg	**.77**	**.77**	**.76**

Three-step sequence tagging

	Precision	Recall	F1	support
O	.84	.81	.83	621
Act	.40	.31	.35	64
Process	.39	.64	.49	42
Attitude + action	.36	.54	.43	112
Attitude	.00	.00	.00	16
Innate quality	.68	.33	.45	69
Learned quality	.76	.73	.75	56
Accuracy			**.69**	980
Macro avg	.49	.48	.47	980
Weighted avg	**.71**	**.69**	**.69**	980

Table 19: Performance evaluation scores of the two approaches. On the left: one-step sequence tagging. On the right: three-step sequence tagging.

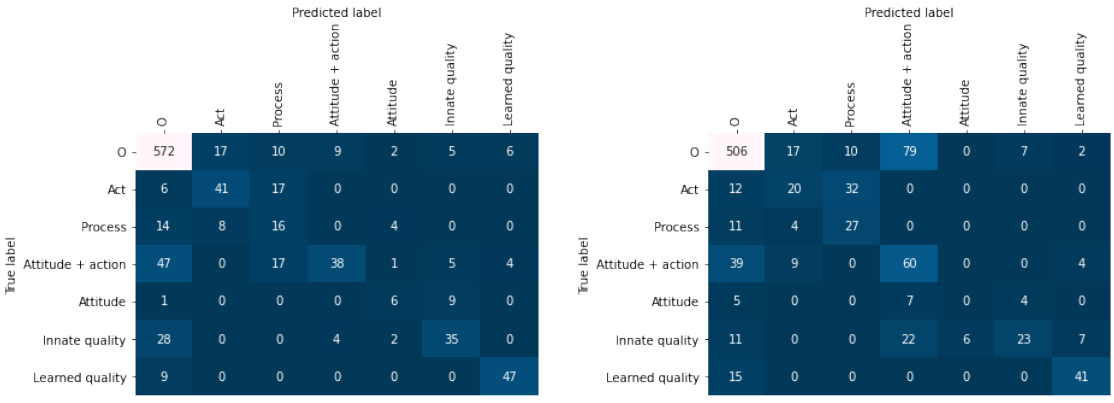


Figure 29: Confusion matrices of the two approaches. On the left: one-step sequence tagging. On the right: three-step sequence tagging.

The one-step sequence tagging model shows a better general performance in table 19. It has a higher recall for O, Act, Attitude, Innate quality, and Learned quality. Not a single attitude that was found by the three-step sequence tagger was correct. It is remarkable, as observed in the confusion matrix in figure 29, that the three-step sequence tagging model predicted 79 words to be Attitude + action that were actually not part of a predicate. These are spurious entities. On the other hand, it found more correct Attitude + action words than the one-step tagger did.

Both models show that words were missed from being classified. This could be completely missed entities or it could be that the predicted entity boundaries were taken more strictly by both of the taggers than by the human annotators.

One-step sequence tagging

	Type	**Partial**	**Strict**	**Exact**
Correct	25	14	11	14
Incorrect	11	0	25	22
Partial	0	22	0	0
Missed	10	10	10	10
Spurious	**41**	**41**	**41**	**41**
Possible	46	46	46	46
Actual	**77**	**77**	**77**	**77**
Precision	.32	.32	.14	.18
Recall	.54	.54	.24	.3
F1	.41	.41	.18	.23

Three-step sequence tagging

	Type	**Partial**	**Strict**	**Exact**
Correct	22	15	10	15
Incorrect	11	0	23	18
Partial	0	18	0	0
Missed	13	13	13	13
Spurious	4	4	4	4
Possible	46	46	46	46
Actual	**37**	**37**	**37**	**37**
Precision	.59	.65	.27	.41
Recall	.48	.52	.22	.33
F1	.53	.58	.24	.36

Table 20: Nervaluate results for all 46 predicates in the two example texts. On the left: one-step sequence tagging. On the right: three-step sequence tagging.

Table 20 shows that the one-step sequence tagger predicted 77 entities, while the rule-based three-step predictor only found 37 entities. The 'true' number of entities in the annotation is 46. That means that the three-step sequence tagger model is closer to the truth. In fact, the one-step tagger seems to have found 41 spurious entities, while the three-step tagger only found 4 spurious entities. As was already stated in section 6.3.2, the one-step tagger is prone to finding spurious entities and short entities.

7.1.0.1 LCM scores

	Text 1	**Text 2**
LCM score human annotated	2.95	1.89
LCM score predicted one-step	2.64	1.92
LCM score predicted three-step	2.75	1.67

Table 21: LCM abstractness scores

Table 21 shows that both taggers found that Text 1 was more abstract than Text 2.

7.2 Conclusion on sequence tagging in practice

Based on tagging of the two texts, three conclusions can be drawn:

- The performance scores on word-level suggest that the one-step tagging approach gives over-all more accurate tags than the three-step approach. The performance scores, however, should be taken with a grain of salt, because of the small scope of this experiment.
- The found entity boundaries seem to be better predicted with the rule-based predicate predictor used in the three-step approach. It gives fewer and longer predicates. This carries the risk of finding two predicates as one entity and mixing up the labels, which might explain the slightly lower word-level performance scores.
- The LCM abstractness measuring formula as adapted to the new label set in the domain of job ads gives reasonably accurate scores for both taggers. This must be verified with more example texts.

8 Conclusion and discussion

In this thesis, the topic of automating the detection of dispositional and behavioural phrasing was studied. We started with outlining the problem. There is a gap between the employment rate of ethnic majority members and ethnic minority members. Companies are slowly but surely working towards a more diverse group of employees, by committing to DEIB. To attract more members of out-groups, more encouragement might be necessary. To make the job market a more welcoming place towards members of ethnic minorities, measures have to be taken to increase the level of inclusivity. This process starts with writing job advertisements that feel inviting towards all groups in society. To achieve this, we need awareness about how different types of language use may be interpreted differently by members of various social group, such that companies can adjust their job advertisements to have a more neutral tone, without having to change the message they want to bring across.

For example, negative meta-stereotypes have been found to occur frequently in job advertisements. These stereotypes can make people hesitate to apply for positions they are perfectly qualified for, because they expect the company to hold a negative stereotype against them. These meta-stereotypes are most likely hidden in what we detected as Innate qualities. More so, according to the Construal Level Theory, the perceived psychological distance correlates with the extent to which people's thinking about a subject is abstract or concrete. A big psychological distance correlates with abstract language. The contrary also seems to hold, that is, that linguistic abstractness influences the perceived social distance to a subject. Since minority (or out-group) members are, socially, farther away than majority (in-group) members from other majority group members, a more concretely written job advertisement may reduce the perceived social distance and thus can come across as more approachable. Besides that, negative meta-stereotypes are expressed in abstract language as requirements about personal characteristics, so detecting abstract language can also be useful to identify these meta-stereotypes in the text. A measurement for 'abstractness' of text was adapted from the LCM score and proposed as a measure for psychological (social) distance. Moreover, the distinction of concrete and abstract phrasing can be written as respectively behavioural and dispositional phrasing. This was our main motivation to do research on detecting dispositional and behavioural phrasing in job advertisements. The main question we tried to answer was "How can behavioural and dispositional wording be automatically detected in job advertisements?"

The behavioural versus dispositional phrasing can also be placed on respectively a spectrum from concrete to abstract language use, where concrete phrasing contains a detailed, exact, step-wise formulation of the message and abstract phrasing takes a more high-level construal, general, multi-interpretable stance. This spectrum of language use was mapped to sub-classes according to the Linguistic Category Model, which is a model that helps to classify interpersonal phrases on the scale from abstract to concrete. From concrete to abstract, the following classes were found: Act, Process, Attitude + action, Attitude, and Innate quality. Learned quality is an extra class that was added for completeness. It is not abstract, but it does belong to the dispositional class together with Innate quality. An important change made as opposed to the LCM and other subsequent studies was the goal to label full predicates instead of single words. This was changed to find the meaning of the language in relation to its context, as words can be interpreted in various ways depending on the words around them.

Three sub-questions were formulated. The first sub-question was about how to find

the relevant sections in the text to classify as dispositional and behavioural phrasing. The second sub-question was about how to create a protocol for annotating these found relevant sections consistently. To answer these questions, a rule-based predicate segmentation method was proposed and an annotation guide was written. After running several annotation pilots, the final annotation guide gave a Krippendorff's alpha Inter Annotator Agreement score of .77 for predicate-level classification on 100 sentences. This contained both labels for relevance as well as sub-labels for the behavioural (Act and Process) and dispositional (Attitude and Quality) classes. The dataset used for further processing was annotated manually by three annotators, according to this written annotation guide.

The third and final sub-question was about how to automate the process of extracting dispositional and behavioural phrasing in-text. Several attempts were made. The first approach was to label the text in three steps. First, a rule-based predictor had to find the boundaries of predicates in text using POS-tags, dependency tags, and word types. The found predicates with these boundaries were then further processed with two classifiers. The first classifier was a binary classifier to determine if the predicate is relevant at all for the task. The second classifier could be either a binary classifier to determine whether a relevant predicate is dispositional or behavioural, or a multi-class classifier using the more extensive set of sub-labels to determine if the predicate was an Act, Process, Attitude (+ action) or (Innate / Learned) quality. Stacking the predicate predictor, relevance classifier, and label classifier gives an in-text tagging system. The models that were tested to train both classifiers were a Decision Tree, Naïve Bayes, Support Vector Machine, Random Forest, and Gradient Boosting with TF-IDF text representation. Another model was trained on Word2Vec embeddings of the text and used an LSTM. The best performing models were given by BERT and RoBERTa, which were fine-tuned to all multi-class and binary classification tasks.

The usability of the system according to the results for the sub-label classifier is arguable, reaching an accuracy of around .70 for both BERT and RoBERTa. Binary classification is far more reliable. Finding the relevant predicates proves to be possible (with a highest accuracy of .88 by RoBERTa), as well as then determining whether the predicate is dispositional or behavioural (with a highest accuracy of .90 by BERT). The results of both the classifiers and the taggers also shows that the classes Innate and Learned quality are relatively easy to identify, while Attitude (+action) is the hardest class to detect correctly. One goal of this study was to find possibly meta-stereotypical predicates. Since it is known that the stereotypes generally hide in requirements asking for certain Innate qualities, it would be most important to detect the Innate qualities accurately. The ROC and PR curves showed that when treating the task as a binary classification task where Innate qualities are classified against 'the rest of the categories' - where it is already given that the predicate is relevant - , this could give an average precision of up to .74 when using BERT. The area under the ROC curve was .93 when using BERT. This means that the chance of being able to distinguish between an Innate quality and NOT Innate quality correctly was 93%.

The second way of automating the process of extracting dispositional and behavioural phrasing was taking a one-step sequence tagging approach. The original sentences were used to fine-tune BERT on a token-classification task for all sub-labels. Relevance was also found directly, because the model tags an entity only if it fits with a relevant label. The accuracy found for this task was .76 on word-level sequence tagging. For this task, similar to the classification task, the Innate and Learned qualities gave the best predictions for the relevant labels, with a recall of respectively .72 and .77.

The two tagging approaches were compared by running the sequence tagging task on two full job advertisement texts. The text was also manually annotated. Although two texts are not suitable ground for drawing reliable conclusions from, the given labeling was examined to find possible suggestions. It was found that the one-step tagger struggles to detect the boundaries of the predicates. It comes up with multiple tags for one predicate. It tags words more spuriously. There are strong indications to believe that a rule-based predicate tagger is more accurate when it comes to predicting predicate boundaries, even if the text is ungrammatical (with some manual domain-specific rules). However, the one-step tagger showed slightly better performance scores for word-level labeling accuracy. These observations show that the models are evenly matched, and it ultimately depends on what the user finds important. Both models found an LCM 'abstractness' score that was higher for the first text than for the second text, which meant that both scores found were in agreement with the LCM score according to the manually made annotations.

8.1 Suggestions for practical applications

Using the model as a dispositional/behavioural language detector, it is applicable in the following ways:

- Visualizing the tags by showing them in-text. At this point, not all classes are detected accurately enough to directly show to users of a writing platform. However, for some classes this might already be possible (especially Innate and Learned qualities).
- Computing a general abstractness score for documents using a formula that weights the found tags. At this point, this is possible, although more testing is advised when using the adapted LCM formula that was provided in this thesis.
- Do meta-analysis such as:

  Comparing abstractness between single documents

  Comparing abstractness between categories of documents

8.2 Discussion and limitations

Multiple problems were encountered during the process of writing this thesis. First of all, there is a class imbalance in the data. Innate qualities have the biggest set of samples and the Attitude labels have the smallest. It was found that the least represented classes are also the least accurately detected classes. Although this is not the only possible explanation for the difference in performance between these classes, an improvement is expected if the models were trained with more data for the under-represented classes. An alternative explanation for this observation could be that certain labels seem to have more variability over the job advertisements. The class Act is highly job-dependent, since the work-related activities are varying over jobs. Qualities, however, are expected to be more widely used. Every employer in the end wants 'reliable' or 'positive' employees with some sort of 'diploma', so the feature set of Qualities is expected to be smaller and thus easier to learn.

In the end, language models are highly dependent on their input data. If the data is highly imbalanced, or otherwise said, does not contain enough diversity, then the model will not be able to learn optimally. We managed to reach up to an accuracy of around .70 for classifying predicates by the relevant sub-labels, which shows that even

with only 2,850 samples of relevant predicates and 6 labels, a classifier can produce decent predictions. The Krippendorff's alpha found for Inter-Annotator Agreement on predicate-level labeling was .77. This is supposably the limit for the accuracy, and so there is still room for improvement. It has to be said, though, that this all holds for the domain of job advertisement texts. It is unknown how well it can be applied to other domains.

Some annotations are more reliable than others. It was seen that the manual annotation strategy is error-prone. Not all annotators agree on all labels, especially considering the large number of edge cases at the side of behavioural classes. This is a problem that the confusion matrix that was made for finding the IAA score already showed. Another limitation of this research is that the rule-based predicate boundary predictions are not fully consistent, foremost caused by the fact that the texts are largely ungrammatical. Ungrammatical text is not guaranteed to be parsed correctly. This has been corrected for partially, by manually adding words and extra rules. Still, this will not have made the automatically found predicate boundaries completely stable.

8.3 Future research

Based on this thesis, one might consider trying the following changes in approach or ideas for follow-up studies:

- Random sentences were selected from the middle of the document (with a deviation) to build the dataset. There might be a better place in the text to sample from. For example, behavioural phrases might generally occur earlier in the text than dispositional phrases. To make up for the data imbalance, it could be preferred to sample from a different place in the text.
- Trying different methods for finding the relevant predicates of the text that are more stable for ungrammatical sentences.
- Even though the annotation guide was improved by running multiple pilots, it was noticeable when looking at incorrect tags that sometimes the given annotation was in fact not right while the predicted tag was. This shows that the annotation task is still difficult. The guide can probably be further improved, for which it would be useful to get input of psychologists or sociologists. It would also be interesting to make use of a semi-automatic labeling - human in the loop - strategy. This can speed up the annotation process.
- This project used Dutch job advertisements. It would be useful for wider application to extend the models to texts in other languages. In English, more tools are already available for associating words to LCM categories (like LIWC). These can be used additionally to boost the performance. One can, vice versa, also consider translating existing tools to Dutch, to compare to results found in this project.
- Make suggestions for alternative formulations of predicates. This was outside of the scope of this project. It might be possible to, for example, through predicate embeddings, find a way to map dispositional phrasing to behavioural phrasing while keeping the intention of the message. This would be an interesting topic to investigate in future research.

Finally, this thesis laid a foundation for doing further social research on the effect of dispositional and behavioural phrasing of job advertisements. An experimental study, using the proposed methods and annotation procedure, could investigate the effect that these types of phrasing have on applicants and/or explore their effects in other domains.

References

Beukeboom, C. J., Tanis, M., & Vermeulen, I. E. (2013). The language of extraversion: Extraverted people talk more abstractly, introverts are more concrete. *Journal of Language and Social Psychology*, *32*(2), 191–201. https://doi.org/10.1177/0261927X12460844

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media, Inc.

Boguslav, M., & Cohen, K. B. (2017). Inter-annotator agreement and the upper limit on machine performance: Evidence from biomedical natural language processing. *Studies in health technology and informatics*, *245*, 298–302.

Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., & Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *CoRR*, *abs/1607.06520*. http://arxiv.org/abs/1607.06520

Caliskan, A., Bryson, J. J., & Narayanan, A. (2016). Semantics derived automatically from language corpora necessarily contain human biases. *CoRR*, *abs/1608.07187*. http://arxiv.org/abs/1608.07187

Coenen, L. H. M., Hedebouw, L., & Semin, G. R. (2006). Measuring language abstraction: The Linguistic Category Model (LCM) manual. Retrieved October 15, 2021, from https://www.yumpu.com/en/document/read/8333454/the-linguistic-category-model-communication-social-cognition-

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. http://arxiv.org/abs/1810.04805

Diversiteit in Bedrijf. (2018). Diversiteitswijzer: Van culturele diversiteit naar inclusie. Retrieved May 8, 2021, from https://issuu.com/diversiteitinbedrijf/docs/diversiteitswijzer_cultuur_-_12-12_

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, *115*(16), E3635–E3644. https://doi.org/10.1073/pnas.1720347115

Hagendoom, L., & Hraba, J. (1987). Social distance toward Holland's minorities: Discrimination against and among ethnic outgroups. *Ethnic and Racial Studies*, *10*(3), 317–333. https://doi.org/https://www.tandfonline.com/action/showCitFormats?doi=10.1080/01419870.1987.9993571

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, *1*(1), 77–89. https://doi.org/10.1080/19312450709336664

Hofstede, G. (2021). Geert Hofstede and Gert Jan Hofstede on culture. Retrieved December 6, 2021, from https://geerthofstede.com/

Hofstede Insights. (2021). *Hofstede Insights Organisational Culture Consulting*. Retrieved December 6, 2021, from https://www.hofstede-insights.com

Honnibal, M., & Montani, I. (2017). *SpaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing* [To appear].

Hutchinson, B., Prabhakaran, V., Denton, E., Webster, K., Zhong, Y., & Denuyl, S. (2020). Social biases in NLP models as barriers for persons with disabilities. *CoRR*, *abs/2005.00813*. https://arxiv.org/abs/2005.00813

Johnson-Grey, K. M., Boghrati, R., Wakslak, C. J., & Dehghani, M. (2020). Measuring abstract mind-sets through syntax: Automating the Linguistic Category Model. *Social Psychological and Personality Science*, *11*(2), 217–225. https://doi.org/10.1177/1948550619848004

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, *abs/1907.11692*. http://arxiv.org/abs/1907.11692

Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Magee, J. C., & Smith, P. K. (2013). The social distance theory of power. *Personality and Social Psychology Review*, *17*(2), 158–186. https://doi.org/10.1177/1088868312472732

Manzini, T., Lim, Y. C., Tsvetkov, Y., & Black, A. W. (2019). Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *CoRR*, *abs/1904.04047*. http://arxiv.org/abs/1904.04047

Nakatani, S. (2014). Language-detection. https://github.com/shuyo/language-detection

Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., & Liang, X. (2018). Doccano: Text annotation tool for human. https://github.com/doccano/doccano

Ningrum, P. K., Pansombut, T., & Ueranantasun, A. (2020). Text mining of online job advertisements to identify direct discrimination during job hunting process: A case study in Indonesia. *PLoS ONE*, *15*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prabhakaran, V., Hutchinson, B., & Mitchell, M. (2019). Perturbation sensitivity analysis to detect unintended model biases. *CoRR*, *abs/1910.04210*. http://arxiv.org/abs/1910.04210

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *CoRR*, *abs/1602.04938*. http://arxiv.org/abs/1602.04938

Schellekens, G. A. C., Verlegh, P. W. J., & Smidts, A. (2010). Language abstraction in word of mouth. *Journal of Consumer Research*, *37*(2), 207–223. https://doi.org/10.1086/651240

Seih, Y., Beier, S., & Pennebaker, J. W. (2017). Development and examination of the Linguistic Category Model in a computerized text analysis method. *Journal of*

Language and Social Psychology, *36*(3), 343–355. https://doi.org/10.1177/0261927X16657855

Semin, G., & Fiedler, K. (1991). The Linguistic Category Model, its bases, applications and range. *European Review of Social Psychology*, *2*(1), 1–30. https://doi.org/10.1177/1948550619848004

Smith, P. (2014). The social distance theory of power. *Youtube*. Retrieved September 6, 2021, from https://youtu.be/KNORJQRhZMM

Snefjella, B., & Kuperman, V. (2015). Concreteness and psychological distance in natural language use. *Psychological Science*, *26*(9), 1449–1460. https://doi.org/10.1177/0956797615591771

Swagerman, A. (2020). CBS jaarrapport integratie 2020. https://longreads.cbs.nl/integratie-2020/

Tulkens, S., Emmery, C., & Daelemans, W. (2016). Evaluating unsupervised dutch word embeddings as a linguistic resource. *CoRR*, *abs/1607.00225*. http://arxiv.org/abs/1607.00225

van den Bosch, A., Busser, B., S.Canisius, & Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. *Computational Linguistics in the Netherlands: Selected Papers from the Seventeenth CLIN Meeting*, 99–114.

Vorauer, J. D., Hunter, A. J., Main, K. J., & Roy, S. A. (2000). Meta-stereotype activation: Evidence from indirect measures for specific evaluative concerns experienced by members of dominant groups in intergroup interaction. *Pers Soc Psychol*, *78*(4), 690–707. https://doi.org/10.1037//0022-3514.78.4.690

Wakslak, C. J., Smith, P. K., & Han, A. (2014). Using abstract language signals power. *Journal of Personality and Social Psychology*, *107*(10), 41–55.

Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to use t-SNE effectively. *Distill*, *1*(10). Retrieved December 10, 2021, from https://distill.pub/2016/misread-tsne/

Wille, L., & Derous, E. (2017). Getting the words right: When wording of job ads affects ethnic minorities' application decisions. *Management Communication Quarterly*, *31*(4), 533–558. https://doi.org/10.1177/0893318917699885

Witkamp, B., Klaver, J., & Timmerman, J. (2018). Het charter diversiteit. https://www.regioplan.nl/wp-content/uploads/2018/10/16005-Eindrapport-Charter-Diversiteit-Regioplan.pdf

Zhao, J., Wang, T., Yatskar, M., Cotterell, R., Ordonez, V., & Chang, K. (2019). Gender bias in contextualized word embeddings. *CoRR*, *abs/1904.03310*. http://arxiv.org/abs/1904.03310

9 Appendix

A Annotation guide final version

Annotatie-handleiding

Beste lezer,

Je gaat zinsdelen uit vacatureteksten annoteren op basis van verschillende labels.

Er zijn de volgende labels:

1. **Een handeling**
2. **Een proces**
3. **Een houding (+ actie)**
4. **Een kwaliteit**
    - **Aangeboren**
    - **Aangeleerd**

Wat niet binnen één van bovenstaande labels past is **niet relevant** of een **onjuist zinsdeel**

De taak wordt verder uitgelegd op de volgende pagina's. Mocht je ergens twijfelen of vragen over hebben dan help ik je graag verder (mail: jetske@textmetrics.com).

Groetjes en succes!

Jetske

Handelingen en processen beschrijven wat de persoon moet **doén** in het werk, dus de inhoud van de werkzaamheden en taken. Dit worden ook wel acties genoemd.

1: Handeling

Dit zijn zinsdelen die **kortdurige** acties beschrijven met een **duidelijk begin en einde**. De actie kan je **gemakkelijk en eenduidig interpreteren en visualiseren** gegeven de context.

Voorbeelden:
“Onder de werkzaamheden valt het geven van presentaties. ”
“Je kunt zowel van tekening lezen als slijpen en lassen. ”
“Het ontvangen en bedienen van gasten is één van de taken. ”
“De volgende taken: monitoren en bijstellen van processen, kalibratie en onderhouden van apparatuur, fermentatie. ”
“Schoonmaken is tenslotte een vak!”

2: Proces

Dit zijn zinsdelen die acties beschrijven met per definitie een **begin en einde**, al kan er **langere tijd** overheen gaan. Ze zijn **moeilijker te visualiseren** omdat ze **multi-interpretabel** zijn: processen bestaan uit een aaneenschakeling van handelingen, dus wat ze precies inhouden hangt af van de situatie.

Voorbeelden:
“Je houdt je bezig met het organiseren van acties. ”
“Je voert de werkzaamheden uit volgens vastgestelde methoden. ”
“Je kunt goed doorvragen naar onderliggende problemen. ”
“Ben jij de collega die een goed feestje kan bouwen achter de bar? ”
“Ook het onderhandelen over contracten behoort tot de dagelijkse werkzaamheden. ”
“Je houd je bezig met het opbouwen en onderhouden van relaties met klanten. ”
“Zo help je de omzetdoelstellingen te realiseren. ”

Houdingen en kwaliteiten gaan over de persoon, dus wie de persoon **ís**. Hieronder vallen kwalificaties, kennis, vaardigheden, interesses en gedrag.

3: Houding

Deze zinsdelen beschrijven een mentale, **langdurige staat van zijn** of **houding** en hebben daarom **geen begin of einde**.

De houding is ofwel een **emotionele consequentie** van een actie of gebeurtenis (*verbaasd, geschrokken*, etc.), ofwel het is een **cognitieve staat** (*denken, begrijpen, betwijfelen, weten*, *wensen,* etc.) of **emotionele staat** (*haten, vrede hebben met, interesseren in*, etc.).

Voorbeelden:
“Je interesseert je in dergelijke problematiek. ”
“Wil je snel aan de slag? ”
“Je houdt rekening met de omstandigheden.”

Houding + actie

Een houding wordt in de zinsdelen regelmatig **gecombineerd** met ofwel een **proces** of **handeling**.

Voorbeelden:
“Je bent het gewend om samen te werken met verschillende afdelingen.”
“Gewend zijn aan” leidt een houding in, maar “samenwerken” is een proces
“Je bent niet bang om producten aan te raden. ”
“Niet bang zijn” leidt een houding in, maar “producten aanraden” is een handeling

4: Kwaliteit

Deze zinsdelen gaan over gevraagde algemene gevraagde (**karakter)eigenschappen**, **kennis** en **vaardigheden** die de persoon bezit. Ze staan volledig **los van de context** en blijven **stabiel** over lange tijd. Als tip: ze staan vaak achter een vorm van “zijn” of “hebben”.

Kwaliteiten zijn de onderscheiden als volgt:

- Aangeboren kwaliteit
  Dit zijn kwaliteiten die je **van jezelf** hebt, je hebt er aanleg voor, al zou je er ook in kunnen groeien. Ze zijn open voor interpretatie en daardoor **moeilijk te verifiëren**.

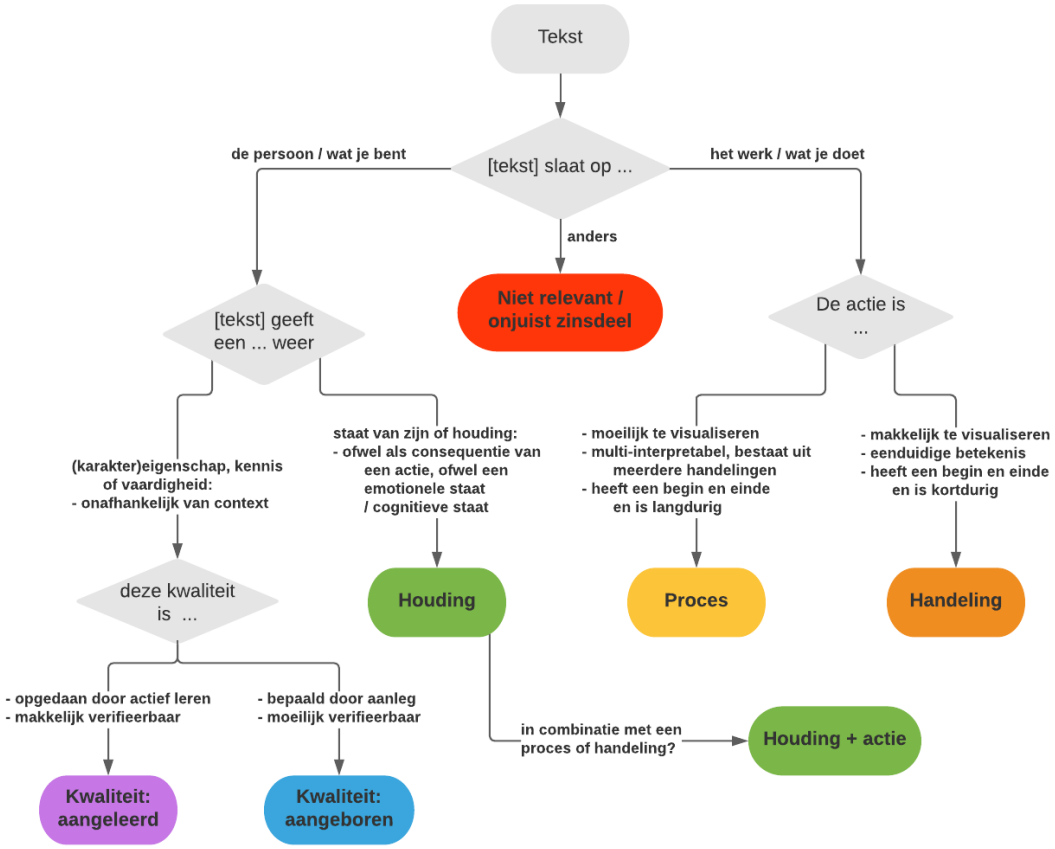
  Voorbeelden:
  “Je bent energiek, gedreven en zelfstandig. ”
  “Je bent zelfstandig, maar ook een teamplayer”
  “Ben jij die enthousiaste collega? ”
  “Er wordt een actieve werkhouding verwacht. ”

- Aangeleerde kwaliteit
  Dit zijn kwaliteiten die je jezelf **actief hebt aangeleerd** in een cursus, op school, of in de praktijk. Ze zijn **makkelijk te verifiëren** aan de hand van een test of diploma.

  Voorbeelden:
  “Je hebt professionele software ontwikkelervaring met Java.”
  “Het hebben van een rijbewijs en eigen vervoer is een pré. ”
  “Je hebt HBO werk- en denkniveau. ”
  “Je beheerst de Nederlandse en Engelse taal goed. ”

Stappenplan

Volg voor annotatie van elk stuk tekst het volgende schema:



Bij twijfel, lees dan nogmaals de uitleg van de labels en maak de daarbij voor jou meest passende keuze.

Past een tekst niet binnen één van de genoemde labels, dus zegt het niks over de persoon of de werkzaamheden (is het bijvoorbeeld algemene informatie over het bedrijf of slaat het op een andere persoon dan de nieuwe werknemer die het bedrijf zoekt), kies dan voor “Niet relevant”.

Het kan soms gebeuren dat de “to-do” tekst niet de goede tekst lijkt te vangen, bijvoorbeeld als het niet het hele relevante zinsdeel bevat of juist meerdere relevante zinsdelen die tot verschillende categorieën behoren. Kies in dat geval voor “onjuist zinsdeel”.

B Annotation pilots

B.1 Annotation pilot 1

The first attempt at annotating by multiple annotators was done in a small pilot. 10 of the same full-length advertisement texts were provided to be annotated by three annotators. The annotators were given the task to freely select the relevant words in the text and give it one of four labels. It was stated that the relevant words are those that are informative about the requirements of the person or about what the job entails.

In the guide, four categories were explained, summarized in the table below. This table was also given in the first annotation guide to help identify simple phrases quickly. It has a similar format as the table provided in the original paper about the LCM by Semin and Fiedler (1991). At the end of the guide, entire job advertisement texts were given with color-coded annotation labels as examples.

Cheat sheet

	Label	**Duration**	**Example**	**Word type**
Act	ACT	Concrete acts, can be done in an afternoon	*Printing* *Cleaning* *Transporting*	Verb
Process	PRO	Multiple acts, can take days/weeks	*Organizing* *Realizing* *Carrying out*	Verb
Attitude	ATT	Constant (mental) state	*Expecting* *Posessing* *Thinking*	Verb
Quality	QUA	Stable characteristics, describing a personality or action	*Independent* *Flexible* *Energetic*	Noun, adjective, or adverb

Table 22: Summary Table

The annotations found were compared and gave a Krippendorff's alpha IAA of 0.43. Additionally, Cohen's kappa was 0.34 between annotator 1 and 2, 0.33 between annotator 2 and 3, and 0.28 between annotator 1 and 3. This means that the resulting annotations were insufficiently consistent to work with.

As can be read from the confusion tables in figure 30, most inconsistency occurs in the top and left row. Those numbers increase when one annotator labels a word that the other annotator does not label. So the annotators label either too much or too little text. One explanation could be that words that should be labeled are overlooked. Another explanation could be that it is unclear what part of the surrounding text of a verb/adjective should be labeled along with it to provide the necessary context.

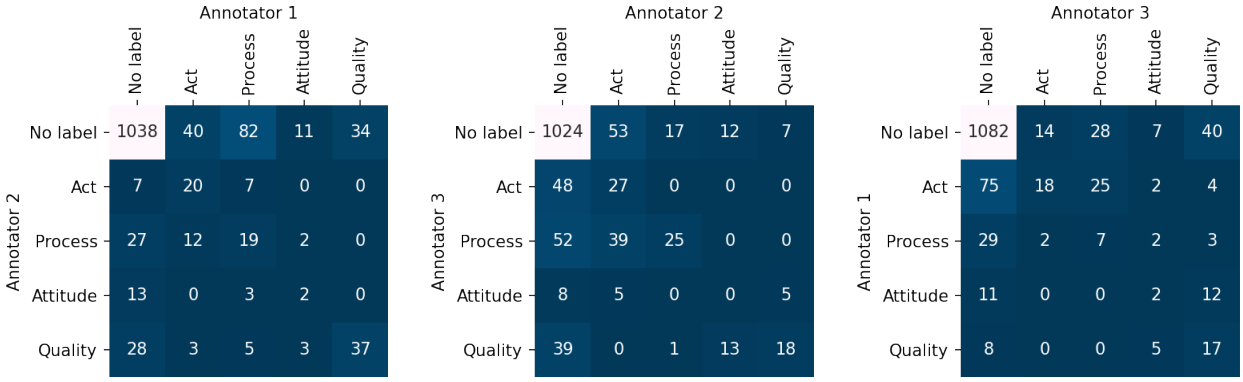


Figure 30: Confusion tables between annotators on word level, including unlabeled words

For each label separately and all labels at once, the IAA scores were computed as given in table 23.

Label	**Krippendorff's Alpha**
"Quality"	0.38
"Attitude"	0.05
"Process "	0.19
"Act"	0.24
All labels	0.43

Table 23: IAA scores for pilot one given by Krippendorff's alpha

Another striking effect is seen in Attitude, which has by far the lowest agreement scores. Looking at the confusion table, it becomes clear that this label is most often confused with Quality. When searching back through the actual annotations, it seems that this is a case of confusion that can be further clarified. To give an example: sentences in the form of *Je bent [energiek/verantwoordelijk/...]* English: 'You are [energetic/responsible/...]' are, understandably, often seen as containing an Attitude (based on the verb "are") while they fall into the definition of Quality. Namely, only the part after "are" should be seen as Quality. Also, incompletely formulated sentences like - *besluitvaardig* English: '- decisive' that need to be filled in imaginatively to - *je bent besluitvaardig* English: '- you are decisive', are also regularly confused as containing a verb and thus incorrectly labeled as Attitude.

B.2 Annotation pilot 2

Two annotators participated in pilot 2.

The following changes were made to the annotation task:

- The texts were split into smaller chunks that would be of more manageable size for annotation. This was done by joining one to three sentences until the resulting string was between 60 and 200 characters. Then, it was said to be suitable for annotation and added to a set of small texts.

- The texts that had to be annotated were filtered beforehand manually based on relevance. They must be on the topic of the content of the job or required characteristics of the applicant. To get there, 50 texts out of the set of short texts were randomly selected, judged on relevance and saved to the annotation set. This process was best to do by hand since the relevance is thought to be hard to interpret automatically. Of course, within every text there are always relevant and irrelevant parts to be found as well.

  The pilot was simplified compared to pilot 1, as the texts were selected by hand taking into account both the length and the relevancy. Example texts from the annotation set are:

  - "Het is verantwoordelijk en zelfstandig werk waarbij je onderdeel bent van een team collega's."
  - "- Besluitvaardigheid. - Mondelinge en schriftelijke communicatie. - Voortgangsbewaking."

- Edge cases were clarified:

  - A clearer distinction was made between Process and Act.
  - A clearer distinction was made between Attitude and Quality.
  - It was stated not to annotate quantities.
  - It was stated not to forget words that are easy to look over like "work".
  - It was stated more clearly when to add additional words to the relevant words:
    if the verb spans over more words, capture it as a whole.
    if the context changes the meaning of the verb, e.g. "building a table" is an Act whereas "building relationships" is a Process.

For pilot 2, the annotations found gave a Krippendorff's alpha of 0.55 and Cohen's kappa of 0.50. When merging the labels Act + Process and Attitude + Quality, the Krippendorff's alpha increases to 0.57 and Cohen's kappa to 0.57 as well. This shows that while showing smaller texts helps to some extent, there is still too much disagreement between the annotators.

Looking at the confusion tables in figure 31, it is even clearer that not the two labels are confused, but that the inconsistency originates in missed cases.

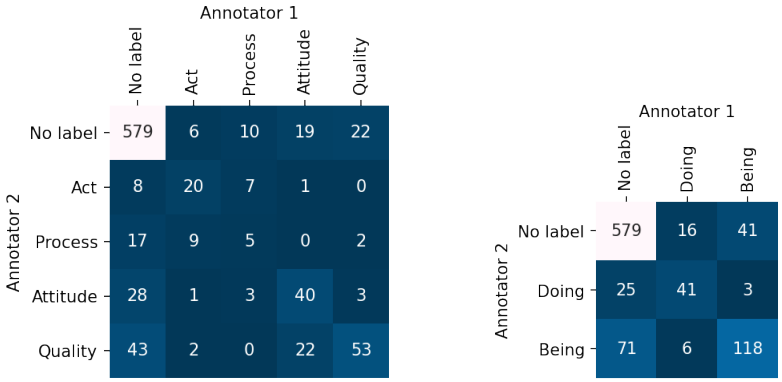


Figure 31: Confusion tables between annotators on word level, including unlabeled words. On the left: all labels. On the right: the merged labels.

Comparing the actual annotations of the two participants, the following stood out:

- It is unclear when nouns and adjectives need to be annotated, e.g.:
  - in *Heb jij iets met planning?* English: 'Do you have affinity with planning?', it is not agreed on whether "planning" is a Quality.
  - in *Heb je iets met technische projecten?* English: 'Do you have affinity with technical projects?', "technical" could be seen as Quality or not. Similarly, in *Je verzorgt projecten binnen kwalitatieve en financiele normen* English: 'You take care of projects within qualitative and financial standards', "qualitative" and "financial standards" could be read as relevant Qualities because they tell about a way to do the job, but should not be labeled as such because they are not about the person but about a work activity. The main verb is "taking care of".
- Quality and Attitude are confused, e.g. in *Je bent in bezit van een rijbewijs* English: 'You have a driver's license' can be seen as an Attitude + Quality or as a whole as Quality.
- General core values of a company could be regarded as separate from personal characteristics, or could be interpreted as qualities that are also expected from employees, e.g. in *Respect en integriteit zijn kernwaarden van de organisatie* English: 'Respect and integrity are core values of the organization'.

B.3 Annotation pilot 3

Two annotators took part in pilot 3.

The annotation guide was revised. The goal was to simplify the task and make it less ambiguous. For this aim, a step-by-step plan was provided to guide the annotators through the process. Similar texts as in pilot 2, selected on length and relevance, were provided. The instructions consisted of two steps:

1. Extract the right sections of the text to annotate, that is, all meaningful verbs and all other words describing personal traits.
2. Determine the right label for each section by walking the section through a flowchart (seen in figure 32).

One consideration made was to tune the original categories as defined in the LCM in such a way that it better fits the goal of the project, thus, to apply it to the idea of dispositional and behavioural language. The behavioural side is linked to concrete language, described with concrete actions and tasks using action verbs (DAV). The dispositional end is linked to abstract language, described with an unexplained characteristic (ADJ). IAV and SAV are, respectively, in between the two on the spectrum. To make sure of this, the annotators were asked to only label Qualities when they referred to a characteristic of a person. At the other extreme, an Act is always defined by a concrete action verb. This was the starting point of building the flowchart. If the text is about a personal trait (stable), it usually leads to a Quality, unless it is written with stable verb like *Je interessert je in ...* English: 'You are interested in ...'. Then it is a stable state or Attitude. Activities and tasks related to the job can never be coded as Quality. These usually end up as a Process or Act, but can in some (rare) cases also lead to Attitude.

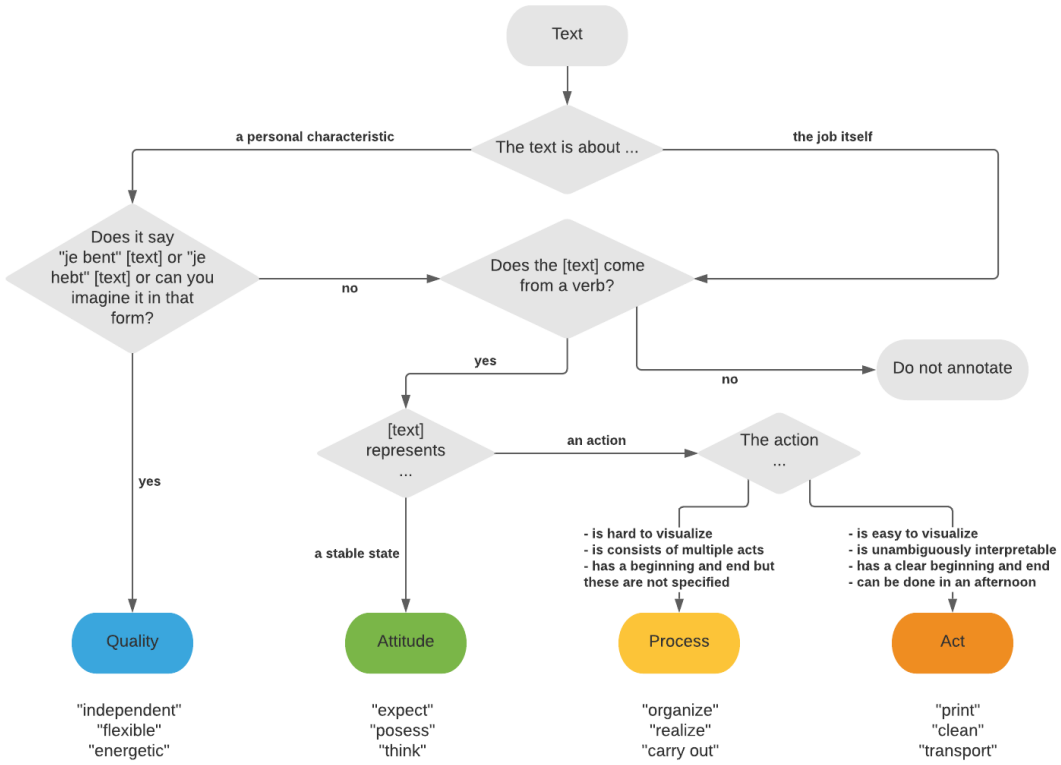


Figure 32: Flowchart given in the annotation guide of pilot 3

A Krippendorff's Alpha IAA score of 0.39 was found. This is very low, and this can be explained by looking at the confusion tables in figure 33.

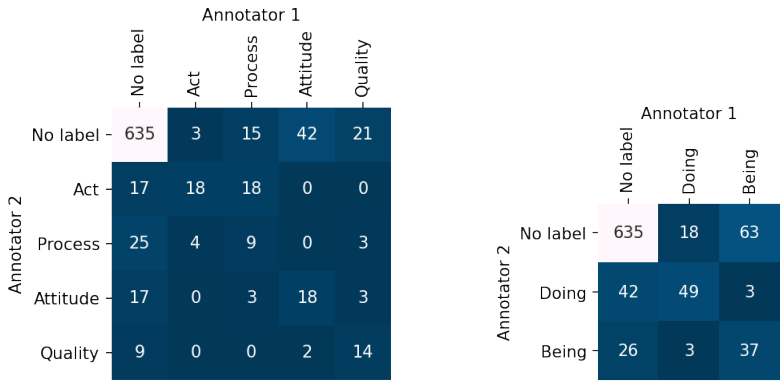


Figure 33: Confusion tables between the annotators on word level, including unlabeled words. On the left: all labels. On the right: the merged labels.

Because of limiting Quality to only capturing personality characteristics, less of these are found than in the pilots before. Labels are mostly confused with a neighboring label,

which is to be expected given that it is a continuous range where there are always edge cases around. It is still striking that there are many missed cases.

The confusion table of the merged labels shows that annotator 1 had 173 words labeled and annotator 2 had 160 words labeled. In total, 241 words were labeled by either one of the annotators. Of these, 92 words were given a label by both of them and 149 were labeled by only one of the annotators. With an overlap of only $92/241 = 0.38$ between the two annotators over all labeled words, a low IAA score is inevitable even if the words labeled by both annotators are high in agreement.

B.4 Annotation pilot 4

Three annotators took part in pilot 4.

To find a way to overcome the problem of missed labels, the annotation task is split up into smaller tasks. Again, 50 short texts were selected to be annotated, based on length and relevance. Annotation took place in three rounds:

1. Annotate only Qualities.
2. Annotate all relevant verbs as "Work activity".
3. Annotate every "Work activity" as either an Attitude, Proces, or Act.

In the first round, the annotators only had one task: annotate Qualities. This should capture the ADJ category of the LCM model. Adjectives are words that are not based on a verb, which makes them stand out from the other categories. As a condition, it was given that these words must always fit after a form of *having* or *being*. For example, by imagining it in a structure such as 'You *are* [initiative/self-reliant/responsible/...]'or 'You *have* [a positive work attitude/affinity with/...]'.

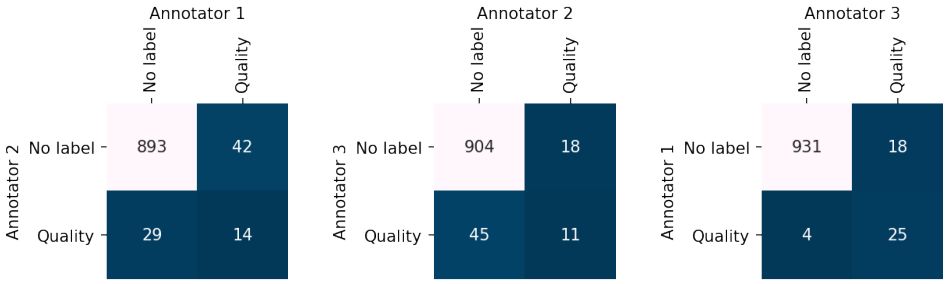


Figure 34: Confusion tables of Quality between annotators

The Cohen's kappa scores were 0.25, 0.68, and 0.23 between the three annotators. Krippendorff's alpha for this task was 0.36. After investigating the words annotated as Quality, the low score is evidently caused by verbs falsely classified as Quality, while they should be an Attitude. These are probably confused because they contain information about a personal trait. The confusion matrices in figure 34 also show that more Qualities are disagreed on than are agreed on. Separating this annotation of this label separately from the other labels might end up giving more confusion rather than less.

In the second round, the annotators had one task: annotate all verbs that inform about either a work related activity, or about an asked person characteristic. This was

done to separate the text selection step from the selection labeling step, such that the steps can be investigated separately.

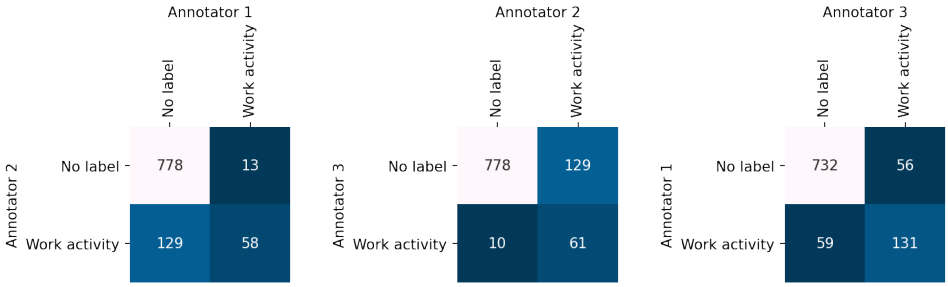


Figure 35: Confusion tables of "Work activity" between annotators

The Krippendorff's alpha found was 0.48. This is very low. From the confusion matrices in figure 35, it is obvious that, for example, the first annotator labels almost (129+58)/(13+58) = 2.6 times as many words as "Work activity" as the second annotator. With such a difference in chosen phrases and/or phrase boundaries, the next step can not give good results.

This is why, for round 3, every annotator was given again the same 50 sentences as for round 2, but with sections already marked as "Work activity". This was done beforehand, to the author's best judgement. This means that the phrase boundaries are a controlled variable and every annotator was provided exactly the same phrases to label as either Attitude, Process, or Act. It was decided to do it this way to create the most ideal, non-ambiguous environment possible for the annotators to determine the right labels.

The results are shown in the confusion tables in figure 36. This task gave Cohen's kappa agreement scores of 0.86, 0.85, and 0.91, and a Krippendorff's alpha score of 0.97.

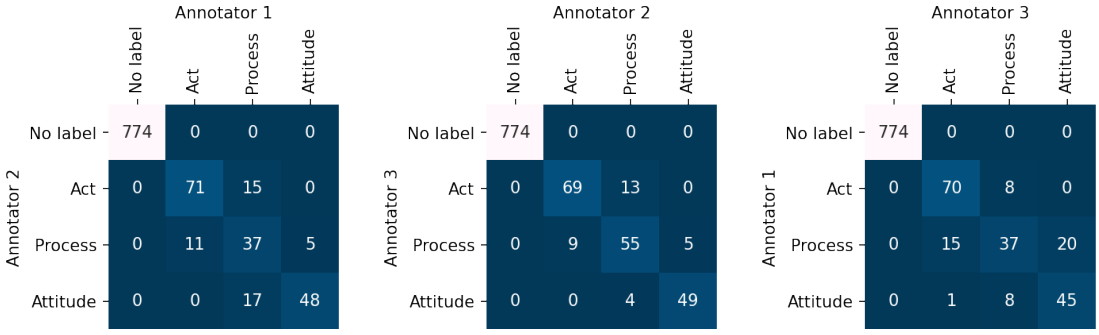


Figure 36: Confusion tables of Acts, Processes, and Attitudes, and unlabeled words between annotators on word level

These results are as expected: most words now seem to be labeled consistently. However, since the to-be labeled text was predefined, this task was now a classification task. Therefore, the words in a phrase together should be treated as one entity. Also, the class "No label" should be left out, as it was stabilized over all annotators. After converting the annotations according to these two changes, a Krippendorff's Alpha of

0.77 was obtained. The Cohen's kappa score between the annotations was respectively 0.60, 0.59, and 0.63. Figure 37 shows that Act and Attitude are confused only once. There is mostly confusion between neighbouring labels.

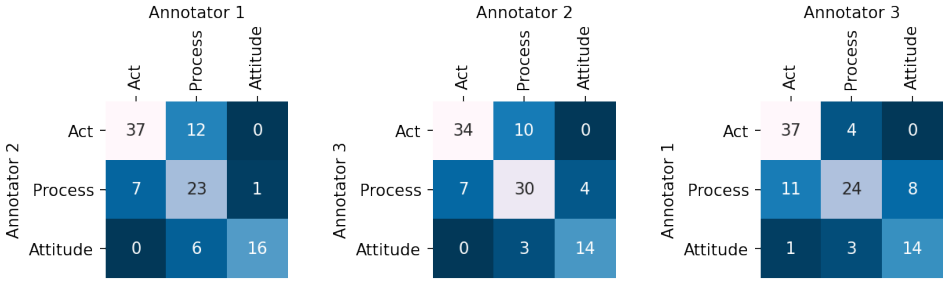


Figure 37: Confusion tables of Acts, Processes, and Attitudes between annotators on phrase level

From this pilot, it was concluded that it is necessary to provide phrase boundaries beforehand to obtain usable annotations for further analysis.

B.5 Annotation pilot 5

Three annotators participated in pilot 5.

B.5.0.1 Predicate segmentation

A new pre-processing step was introduced. Just like how in pilot 4 round 3 the "Work activity" phrase boundaries were selected by hand, boundaries for all 4 labels had to be predefined automatically. It was decided to find predicates as phrases. The automatically found predicates are then provided to the annotation task, changing the annotation task to a classification task.

A large set of random sentences from the dataset was sent through Frog. Frog is a POS tagger that comes with the software distribution "LaMachine" (van den Bosch et al., 2007). It is a tool that integrates NLP modules like tokenization, lemmatization, dependency parsing, base phrase chunking, NER, and morphological segmentation for the Dutch language. This tool provided the tags needed to further process the texts.

As discussed in section 2.4, previous studies used the LCM to label verbs as one of the categories DAV, IAV, and (S)AV, and adjectives, adverbs, and nouns as ADJ. However, this is where the goal deviates from previous pilots and previous research. In pilot 5, it was chosen not to label single words, but to label predicates that usually include some context. This was decided on because the context might in fact change the label given to the text in question. To illustrate this: *bearing responsibility* for a task (SAV) is not the same as being a *responsible* person (ADJ). Responsibility can be related to a job-related activity or to a person characteristic. Therefore, it was decided to find segments in the sentences such that they highlight the predicates belonging to verb phrases. It is hypothesized here that the abstractness of a sentence or text as a whole is better captured in this way than through words alone.

Various tags were combined using rules to find the sentences' predicates. These are the following:

- All predicates are complete verb phrases, so they usually start with a word with tag "B-VP".
- They may also start with the (in)finitive form of a verb, so look for:

  "WW(inf,nom,zonder,zonder-n)"

  "WW(inf,vrij,zonder)"

  "WW(pv"
- Sometimes they can start with a noun, so also look for:

  "N(soort,ev,basis"

  "N(soort,mv,basis"
- Some phrases start with the relevant nouns ["ervaring", "kennis", "beheersing"], so look for those words in specific as well
- A predicate could also start with an adjective, so look for the combination ["ADJ", "vrij", "basis", "zonder"]
- End a predicate when encountering a punctuation mark ".", "!", "?", ";", or ":".
- Sometimes, end the predicate at "en" or ",", namely only if a new predicate beginning is found right after, or if the dependency score after the comma is lower than that for the word in front of the comma - then it likely belongs to a new verb phrase that is introduced later. In enumerations, the words usually have the same relation to the main verb. so they are on the same dependency level. In this case, they are added to the same predicate.

In order to obtain the right tags through Frog, it is important that the sentences are grammatical to begin with. An issue that arised immediately is that many sentences in job advertisements are written in an incomplete way (e.g. '- takes initiative', 'caring and responsible', 'The writing of articles', 'attending internal meetings', 'customer oriented', 'collaboration.', etc.). This is because predicates are usually verbs or verb phrases, and a verb phrase can not be identified without a verb at its basis. The sentences in the annotation set were made grammatical to be (more likely) given the correct POS- and dependency tags by the parser. This was done manually by adding minimal phrases such as 'you are ...' and adjusting the punctuation (e.g. 'customer oriented' → 'You are customer oriented' or 'caring and responsible' → 'We value being caring and responsible').

This resulted in predicates as given in figure 38.

```
Functie-eisen : je [hebt ervaring met elektra en leidingen en aantoonbare ervaring met het monteren van keukens ].
Je [bent communicatief vaardig ].
Je [bent in het bezit van een rijbewijs ]B. Je [bent in het bezit van VCA ].
Je [hebt via cv aantoonbare kennis van strategisch beleid , richtlijnen en procedures op het taak en deelgebied ( 10 % ) ].
Je [hebt zeer goede beheersing van het Microsoft office pakket en een FMIS pakket ( 5 % ) ].
Je [hebt via cv aantoonbare werkervaring met projecten waar het gaat om asbestverwijdering ( 10 % ) ].
Je [hebt ervaring met het opstellen en beheren van een asbestbeheerplan ( 5 % ) ].
De [klas vormt vanaf het begin een orkest en vanuit het samenspel wordt de techniek van de instrumenten aangeleerd ].
Je [onderhoudt duurzame klantrelaties ], [signaleert mogelijke extra werkzaamheden ]en [denkt mee over totaaloplossingen ].
Je [verzorgt de calculaties binnen de afdeling ].
```

Figure 38: Random selection of sentences with the automatically found predicate boundaries given.

B.5.0.2 Final changes to the annotation guide

Lastly, the annotation guide was changed such that a clearer separation is made between *being* and *doing* to help annotators distinguish actions (DAV and IAV categories) from stable characteristics (ADJ and SAVs). Along with a new flowchart came short explanations with examples given for each possible label.

Four labels were added:

1. A label "Not relevant" was added for predicates that do not fit with any of the categories. This could happen, because random sentences were given and not all sentences of a job advertisement contain relevant information for this task. Some contain more general information about the company, for example.

2. A label "Incorrect predicate" was added for incorrectly found predicate boundaries. This could occur, because the automatically found rule-based predicate boundaries are not flawless and can include predicates where part of the relevant text is outside of the left or right boundary even though the predicate itself is relevant for annotation. On the other hand, an automatically found predicate may contain two predicates that should have been found separately.

3. The label Quality was subdivided into Innate and Learned qualities. This was not originally part of the LCM, but the distinction is crucial to make. Generally, Innate qualities are those that are generalizing and seen as 'non-inclusive'. These are words like "responsible", "active", "enthusiastic". On the other hand, Learned qualities tell about someone's qualifications and skills that can be tested or measured fairly well. For example: 'You master the English language' or 'You have a driver's license'. Such requirements are not up for debate and also not questioned as being discriminative.

4. One sub-label was added for Attitude, namely Attitude + action. This can be used in case a predicate contains two verbs, where the main verb introduces a state and the other verb reveals an action that is related to the state. It describes how someone feels about performing some action or as a consequence of an action, rather than stating directly that the action is part of the job. For example: 'You are not scared to help customers'. Without this label, the annotator might be confused about whether to code this as an Attitude or as an action.

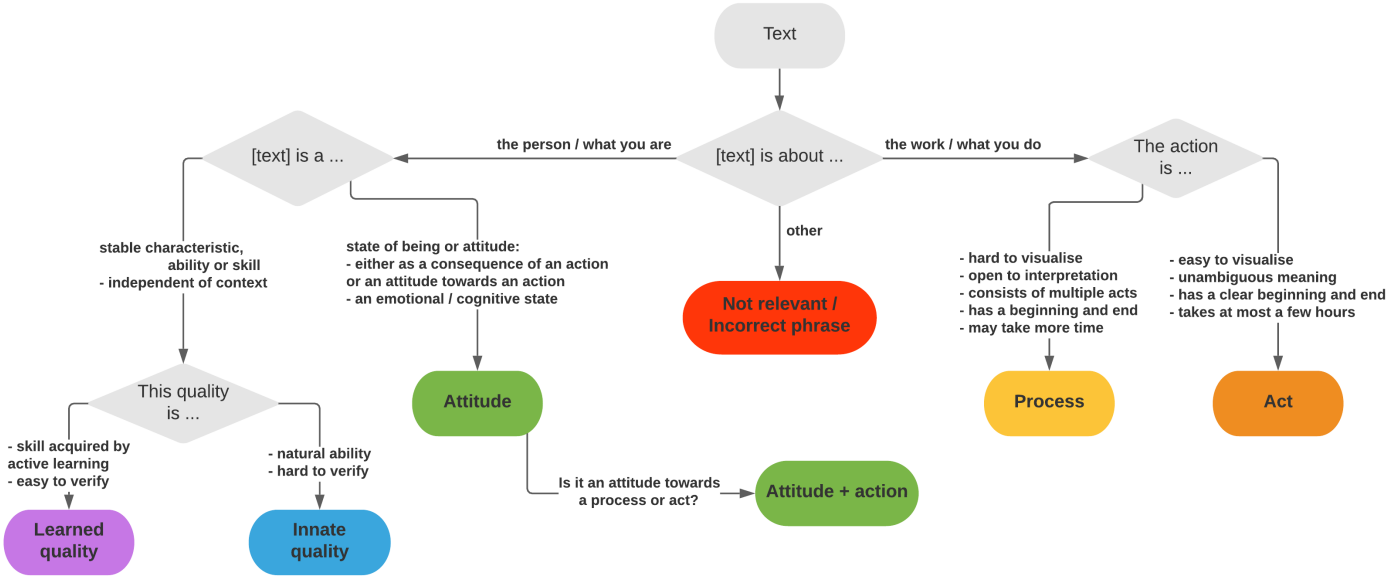


Figure 39: Flowchart annotation guide pilot 5

Additionally, the flowchart was updated as seen in figure 39.

B.5.0.3 Results

This simpler way to annotate the data with the predicate boundaries defined beforehand is expected to result in more reliable annotations, because there can only be confusion about what label to give each text predicate. In pilot 4, this way of providing predefined boundaries already gave a high agreement score already. However, it had to be checked again because three changes were made since then:

1. The predicate boundaries are defined as containiing full predicates instead of verbs (with little context) only.
2. The boundaries for the predicates were found automatically through their POS tags instead of by hand. These automatically found predicates are not flawless (although neither are hand-made ones).
3. Extra (sub-)labels were added to obtain more complete information out of the data.

Each of the three annotators were given the same 100 sentences with the automatically found predicate boundaries.

While this task was a classification task, the end goal is still a sequence labeling task. Therefore, IAA can be measured in two ways:

- Comparing the annotations on word level, including the parts of the sentences that are not part of a detected predicate. This gave a Krippendorff's alpha of 0.81.

- Comparing the annotations on predicate level, leaving out the text outside of the predicate boundaries and treating the words inside the predicates as single entities. This gave a Krippendorff's alpha of 0.71.

The latter is more informative when it comes to evaluating the quality of the annotators' judgements rather than the quality of the predicate boundaries. This is analyzed further below.

The confusion tables show the following:

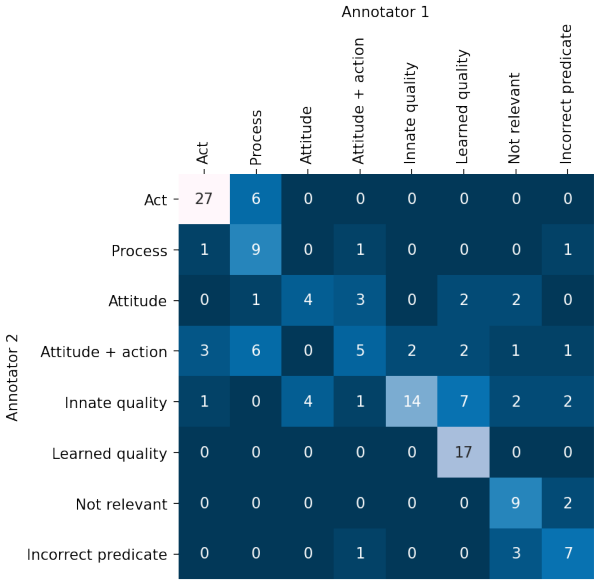


Figure 40: Confusion table of all labels between annotator 1 and 2

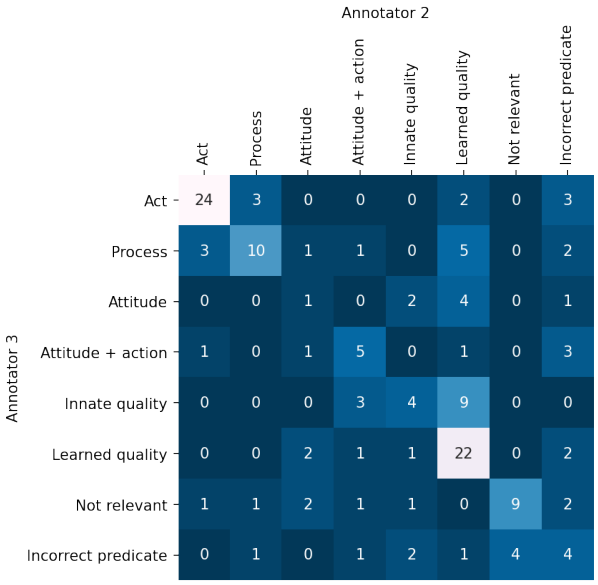


Figure 41: Confusion table of all labels between annotator 2 and 3

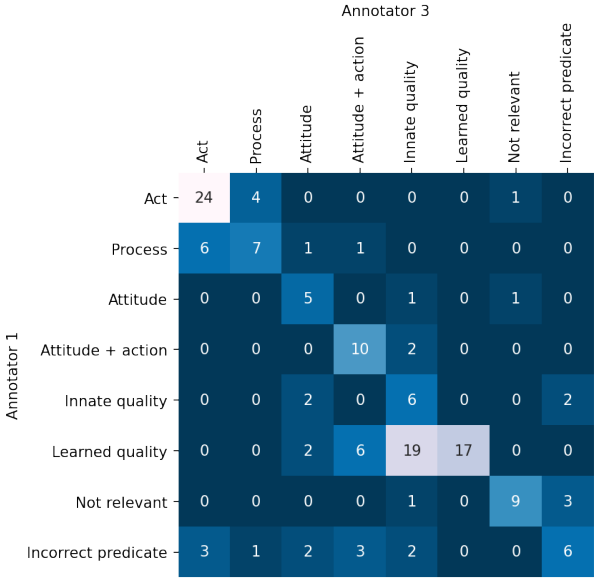


Figure 42: Confusion table of all labels between annotator 3 and 1

The least confusion is found in the labels Act and Innate quality. This is expected, because these are found on the extreme ends of the abstract-concrete scale. It is remarkable that Innate and Learned qualities seem to be confused by one annotator. Most confusion can be seen to occur in determining whether the predicate boundaries are correct or not, so the annotators were not equally strict in this decision.

From the confusion tables and IAA scores it was concluded that this way of annotation gave sufficient agreement to continue with creating the actual annotated dataset.

C Applied detection in practice

C.1 Text 1:

C.1.1 Manually annotated text

Als Helpende vervul je een essentiële rol in het dagelijks leven van zorgbehoevenden **Process** .

Jij bent de persoon die ervoor zorgt dat jouw cliënt zijn of haar dag goed begint en/ of deze in de avond prettig afsluit **Attitude + action** .

Jouw aanwezigheid maakt daarom een wereld van verschil.

Je takenpakket is gevarieerd en omvat het ondersteunen van cliënten **Process** verschillende dagelijkse activiteiten.

Dit kan betekenen dat je ze helpt zichzelf te wassen, aan te kleden **Act** en het eventueel assisteert bij het aantrekken van steunkousen **Act** .

Daarnaast zorg je ervoor dat de cliënt zijn of haar medicatie tijdig toegediend krijgt **Act** .

In de zorg draait alles om mensenwerk.

Het is dan ook enorm belangrijk dat je je in kunt leven in de situatie van een cliënt **Innate quality** .

Kun jij goed luisteren **Innate quality** en vind je het leuk om een gezellig praatje te maken **Attitude + action** tijdens de verzorgende werkzaamheden?

Dan zorgen jouw warmte, oprechte interesse, glimlach en luisterend oor **Innate quality** ervoor dat de cliënt helemaal op zijn of haar gemak voelt.

Als je bijvoorbeeld in een verpleeg- en verzorgingshuis, gehandicaptenzorg of ziekenhuis aan de slag gaat **Process** , kom je in een team te werken met leuke collega' s.

Kies je voor het draaien van diensten in de thuiszorg **Process** , dan bezoek je zelfstandig hulpbehoevenden **Process** tijdens een route bij jou in de buurt.

Ben je enthousiast geworden om als Helpende niveau 2(medicatie) aan het werk te gaan in Nootdorp e. o. en voldoe je aan de funtie- eisen?

Welkom!

Schrijf jezelf veilig online in.

Requirements » Je hebt een afgeronde opleiding helpende niveau 2 **Learned quality** .

» Een afgeronde medicatiemodule **Learned quality** .

» Je bent bereid om een Verklaring Omtrent het Gedrag aan te vragen **Attitude + action** (voor medewerkers in loondienst vergoeden wij de kosten ná je eerste dienst).

» Je beschikt over een sterk ontwikkeld verantwoordelijkheidsgevoel **Innate quality** .

» Je bent enthousiast om bij één of meerdere opdrachtgevers van Zorgwerk aan het werk te gaan **Attitude + action** .

» Je bent communicatief vaardig **Innate quality** .

» Je hebt een goede beheersing van de Nederlandse taal in zowel woord als geschrift **Learned quality** .

» Je hebt een cliëntgerichte en gastvrije instelling **Innate quality** .

» Je kan zowel zelfstandig als in teamverband werken **Innate quality** .

» Je hebt signalerend en probleemoplossend vermogen **Innate quality** .

» Je bent zeer betrouwbaar **Innate quality** , de opdrachtgevers kunnen op jou rekenen.

» Je komt representatief over voor de zorg, qua kledingkeuze en gedrag **Innate quality** .

» Je hebt affiniteit met Thuiszorg en thuiszorgroute in de wijk **Attitude** .

C.1.2 One-step sequence tagging

Als Helpende vervul je een essentiële rol Attitude in het dagelijks leven van zorgbehoevenden Process .

Jij bent Attitude de persoon die ervoor zorgt dat jouw cliënt zijn Attitude + action of haar dag goed begint Process en/ of deze Process in de avond Attitude + action prettig Process afsluit.

Jouw aanwezigheid maakt daarom een wereld van verschil.

Je takenpakket is gevarieerd en omvat het ondersteunen van cliënten verschillende dagelijkse activiteiten Process .

Dit kan betekenen dat je ze helpt zichzelf te wassen, aan te kleden en het eventueel assisteert bij het aantrekken van steunkousen Act .

Daarnaast zorg je ervoor dat de Process cliënt zijn Act of Process haar Act medicatie tijdig toegediend krijgt Process .

In de zorg draait alles om mensenwerk.

Het is dan ook enorm belangrijk dat Innate quality je je in kunt leven in de situatie van een cliënt.

Kun jij goed luisteren Attitude + action en vind je het leuk om een gezellig praatje te maken tijdens de verzorgende werkzaamheden Attitude + action ?

Dan zorgen jouw warmte, oprechte interesse, glimlach en luisterend Innate quality oor ervoor dat de cliënt helemaal op zijn of haar gemak voelt Attitude .

Als je bijvoorbeeld in een verpleeg- en verzorgingshuis, gehandicaptenzorg of ziekenhuis aan de slag gaat, kom je in een team te werken met leuke collega' s.

Kies je voor het draaien van diensten in de thuiszorg Process , dan bezoek je Act zelfstandig Process hulpbehoevenden Act tijdens Process een route bij jou in de buurt Act .

Ben je enthousiast geworden om als Helpende niveau 2(medicatie) aan het werk te gaan in Nootdorp e. o. en voldoe je aan de funtie- eisen?

Welkom!

Schrijf jezelf veilig online in.

Requirements » Je hebt een afgeronde opleiding helpende niveau Learned quality 2.

» Een afgeronde medicatiemodule.

» Je bent bereid Attitude + action om een Learned quality Verklaring Omtrent het Learned quality Gedrag aan te Attitude + action vragen Learned quality (voor medewerkers in loondienst vergoeden wij de kosten ná je eerste dienst).

» Je beschikt over een sterk ontwikkeld verantwoordelijkheidsgevoel Innate quality .

» Je bent enthousiast om bij één of meerdere opdrachtgevers van Zorgwerk aan het werk te gaan Attitude + action .

» Je bent communicatief vaardig Innate quality .

» Je hebt een goede beheersing van de Nederlandse taal in zowel woord als geschrift Learned quality .

» Je hebt een cliëntgerichte Innate quality en Attitude gastvrije Innate quality instelling Attitude .

» Je kan zowel zelfstandig als in teamverband werken Innate quality .

» Je hebt signalerend en probleemoplossend vermogen Innate quality .

» Je bent zeer betrouwbaar Innate quality , de opdrachtgevers kunnen op jou rekenen Innate quality .

» Je komt representatief over voor de zorg, qua kledingkeuze en gedrag.

» Je hebt affiniteit met Thuiszorg en thuiszorgroute in de wijk Innate quality .

C.1.3 Three-step sequence tagging

Als Helpende vervul je een essentiële rol in het dagelijks leven van zorgbehoevenden Process .

Jij bent de persoon die ervoor zorgt dat jouw cliënt zijn of haar dag goed begint en/ of deze in de avond prettig afsluit Attitude + action .

Jouw aanwezigheid maakt daarom een wereld van verschil.

Je takenpakket is gevarieerd en omvat het ondersteunen van cliënten verschillende dagelijkse activiteiten Process .

Dit kan betekenen dat je ze helpt zichzelf te wassen, aan te kleden en het eventueel assisteert bij het aantrekken van steunkousen Act .

Daarnaast zorg je ervoor dat de cliënt zijn of haar medicatie tijdig toegediend krijgt Process .

In de zorg draait alles om mensenwerk.

Het is dan ook enorm belangrijk dat je je in kunt leven in de situatie van een cliënt Attitude + action .

Kun jij goed luisteren Attitude + action en vind je het leuk om een gezellig praatje te maken tijdens de verzorgende werkzaamheden Attitude + action ?

Dan zorgen jouw warmte, oprechte interesse, glimlach en luisterend oor ervoor dat de cliënt helemaal op zijn of haar gemak voelt Attitude + action .

Als je bijvoorbeeld in een verpleeg- en verzorgingshuis, gehandicaptenzorg of ziekenhuis aan de slag gaat, kom je in een team te werken met leuke collega' s.

Kies je voor het draaien van diensten in de thuiszorg Process , dan bezoek je zelfstandig hulpbehoevenden tijdens een route bij jou in de buurt Act .

Ben je enthousiast geworden om als Helpende niveau 2(medicatie) aan het werk te gaan in Nootdorp Attitude + action e. o. en voldoe je aan de funtie-eisen?

Welkom!

Schrijf jezelf veilig online in.

Requirements » Je hebt een afgeronde opleiding helpende niveau 2 Learned quality .

» Een afgeronde medicatiemodule.

» Je bent bereid om een Verklaring Omtrent het Gedrag aan te vragen(voor medewerkers in loondienst vergoeden wij de kosten ná je eerste dienst) Attitude + action .

» Je beschikt over een sterk ontwikkeld verantwoordelijkheidsgevoel Innate quality .

» Je bent enthousiast om bij één of meerdere opdrachtgevers van Zorgwerk aan het werk te gaan Attitude + action .

» Je bent communicatief vaardig Innate quality .

» Je hebt een goede beheersing van de Nederlandse taal in zowel woord als geschrift Learned quality .

» Je hebt een cliëntgerichte en gastvrije instelling Attitude .

» Je kan zowel zelfstandig als in teamverband werken Learned quality .

» Je hebt signalerend en probleemoplossend vermogen Innate quality .

» Je bent zeer betrouwbaar, de opdrachtgevers kunnen op jou rekenen Innate quality .

» Je komt representatief over voor de zorg, qua kledingkeuze en gedrag.

» Je hebt affiniteit met Thuiszorg Innate quality en thuiszorgroute in de wijk.

C.2 Text 2:

C.2.1 Manually annotated text

Bedrijfsautotechnicus(BAT/ EBAT) Nijmegen.

Transdev- Nijmegen Gisteren Functieomschrijving.

Bedrijfsautotechnicus(BAT/ EBAT) Nijmegen.

Standplaats: Nijmegen I Aantal uur: 40 uur per week I Niveau: MBO.

Hou je van het échte grote werk Attitude ?

Wil je werken aan de nieuwste bussen Attitude + action ?

En wil jij samen met ons de omslag doormaken naar zero emissie Attitude + action ?

Dan ben je bij onze werkplaatsen aan het juiste adres!

Wat ga je doen Ons werk?

Sleutelen aan verschillende voertuigen Process : Bussen(diesel/ elektrisch), touringcars en taxibussen.

We zoeken vakmensen die hun hand niet omdraaien voor onderhoud, reparatie en het verhelpen van storingen Attitude + action .

Die kennis hebben van elektronica Learned quality of zich daarin willen ontwikkelen Attitude + action .

Die accuraat, zelfstandig en klantgericht zijn Innate quality .

Als bedrijfsautotechnicus verricht je diagnoses, reparatie- en onderhoudswerkzaamheden Act .

Je zorgt dat de voertuigen bedrijfszeker zijn Process door een correcte uitvoering van periodiek onderhoud, het correct uitvoeren van alle voorkomende reparaties en het doeltreffend verhelpen van storingen Act .

In de werkplaats voeren we complexe reparaties uit van bussen Act en stellen we diagnoses Act (evt. met behulp van diagnose- apparatuur en- gereedschap).

We werken met diverse merken waardoor jij je continue blijft ontwikkelen Process .

Het werk aan elektrische bussen is schoon in tegenstelling tot diesel voertuigen, omdat de werkzaamheden voornamelijk binnenin plaatsvinden dan onder de bus.

Wat breng je mee?

0- 3 jaar relevante werkervaring Learned quality ;.

Bereidheid tot het volgen van(leveranciers) trainingen Attitude + action ;.

Een afgeronde opleiding BAT of EBAT Learned quality ;.

Rijbewijs B(eventueel C) en D Learned quality (of bereid deze te halen);.

Kennis van moderne elektronica, pneumatiek en hydraulica is een pre Learned quality ;.

Bereidheid om in een rooster te willen werken Attitude + action .

Wat bieden wij?

Diplomatoeslag en jouw eigen gereedschapskist.

Vrijheid en zelfstandigheid in je functie.

Een collegiaal en betrokken team wat voor elkaar klaar staat.

Daarnaast werken wij met diverse grote merken, zoals VDL, BYD, Mercedes, MAN etc. En zijn er diverse mogelijkheden voor verdere ontwikkeling.

Verder bieden wij Een goed salaris conform CAO OV, 8 [UNK] vakantiegeld en een eindejaarsuitkering van 1, 5 [UNK] [UNK] [UNK] 400,- bruto per jaar(dit is exclusief toeslagen, mocht dit van toepassing zijn);.

23 tot 29 vakantiedagen, afhankelijk van leeftijd;.

17 of 26 ATV dagen, afhankelijk van leeftijd;.

Een goede pensioenregeling.

Wie zijn wij?

Transdev Nederland geeft mensen de vrijheid om te reizen.

Als toonaangevend mobiliteitsbedrijf zijn we actief in het hele land onder diverse merknamen zoals Connexxion, Hermes en Witte Kruis.

We leveren innovatieve mobiliteitsoplossingen voor personenvervoer en dragen met onze duurzame voertuigen en initiatieven bij aan de energietransitie in ons land.

Bij alles wat we doen staan onze reizigers en medewerkers centraal.

Transdev Nederland heeft 8. 000 medewerkers en vervoert dagelijks zo' n 500. 000 mensen.

Enthousiast?

Ben je enthousiast geworden om aan de slag te gaan als bedrijfsautotechnicus?

Solliciteer dan snel!

Vergeet niet je CV en motivatiebrief in te sturen.

C.2.2 One-step sequence tagging

Bedrijfsautotechnicus(BAT/ EBAT) Nijmegen.

Transdev- Nijmegen Gisteren Functieomschrijving.

Bedrijfsautotechnicus(BAT/ EBAT) Nijmegen.

Standplaats: Nijmegen I Aantal uur: 40 uur per week I Niveau: MBO.

Hou je van het échte grote werk Attitude ?

Wil je werken aan de nieuwste bussen?

En wil jij samen met ons de omslag doormaken Process naar zero emissie Process ?

Dan ben je bij onze werkplaatsen aan het juiste adres!

Wat ga je doen Ons werk?

Sleutelen aan verschillende voertuigen: Bussen(diesel/ elektrisch), touringcars en taxibussen.

We zoeken vakmensen die hun hand niet omdraaien voor onderhoud, reparatie en het verhelpen van storingen.

Die kennis hebben van elektronica Learned quality of zich daarin willen ontwikkelen.

Die accuraat, zelfstandig en klantgericht zijn Innate quality .

Als bedrijfsautotechnicus verricht Act je Process diagnoses, reparatie- en onderhoudswerkzaamheden Act .

Je zorgt dat de voertuigen bedrijfszeker zijn Act door een correcte Process uitvoering van periodiek onderhoud Act , het correct Process uitvoeren van alle voorkomende reparaties Act en het Process doeltreffend verhelpen van storingen Act .

In de werkplaats voeren we complexe Act reparaties uit van Act bussen en stellen we diagnoses(Act evt. met behulp van diagnose- Act apparatuur en- gereedschap Act).

We werken met diverse merken waardoor jij je continue blijft ontwikkelen.

Het werk aan elektrische bussen is schoon in tegenstelling tot diesel voertuigen, omdat de werkzaamheden voornamelijk binnenin plaatsvinden dan onder de bus.

Wat breng je mee?

0- 3 jaar relevante werkervaring Learned quality ;.

Bereidheid tot het volgen van(leveranciers) trainingen Process ;.

Een afgeronde opleiding BAT of EBAT Learned quality ;.

Rijbewijs B(eventueel C) Learned quality en D Learned quality (of bereid deze te halen) Learned quality ;.

Kennis van moderne elektronica, pneumatiek en hydraulica is een pre Learned quality ;.

Bereidheid om Innate quality in een rooster te Innate quality willen werken Innate quality .

Wat bieden wij?

Diplomatoeslag en jouw eigen gereedschapskist.

Vrijheid en zelfstandigheid in je functie.

Een collegiaal en betrokken team wat voor elkaar klaar staat.

Daarnaast werken wij met diverse grote merken, zoals VDL, BYD, Mercedes, MAN etc. En zijn er diverse mogelijkheden voor verdere ontwikkeling.

Verder bieden wij Een goed salaris conform CAO OV, 8 [UNK] vakantiegeld en een eindejaarsuitkering van 1, 5 [UNK] [UNK] [UNK] 400,- bruto per jaar(dit is exclusief toeslagen, mocht dit van toepassing zijn);.

23 tot 29 vakantiedagen, afhankelijk van leeftijd;.

17 of 26 ATV dagen, afhankelijk van leeftijd;.

Een goede pensioenregeling.

Wie zijn wij?

Transdev Nederland geeft mensen de vrijheid om te reizen.

Als toonaangevend mobiliteitsbedrijf zijn we actief in het hele land onder diverse merknamen zoals Connexxion, Hermes en Witte Kruis.

We leveren innovatieve mobiliteitsoplossingen voor personenvervoer en dragen met onze duurzame voertuigen en initiatieven bij Process aan de energietransitie in ons land.

Bij alles wat we doen staan onze reizigers en medewerkers centraal.

Transdev Nederland heeft 8. 000 medewerkers en vervoert dagelijks zo' n 500. 000 mensen.

Enthousiast?

Ben je enthousiast geworden om aan de slag te gaan Attitude + action als bedrijfsautotechnicus?

Solliciteer dan snel!

Vergeet niet je CV en motivatiebrief in te sturen.

C.2.3 Three-step sequence tagging

Bedrijfsautotechnicus(BAT/ EBAT) Nijmegen.

Transdev- Nijmegen Gisteren Functieomschrijving.

Bedrijfsautotechnicus(BAT/ EBAT) Nijmegen.

Standplaats: Nijmegen I Aantal uur: 40 uur per week I Niveau: MBO Learned quality .

Hou je van het échte grote werk Attitude + action ?

Wil je werken aan de nieuwste bussen?

En wil jij samen met ons de omslag doormaken naar zero emissie?

Dan ben je bij onze werkplaatsen aan het juiste adres!

Wat ga je doen Ons werk?

Sleutelen aan verschillende voertuigen: Bussen(diesel/ elektrisch), touringcars en taxibussen.

We zoeken vakmensen die hun hand niet omdraaien voor onderhoud, reparatie en het verhelpen van storingen Act .

Die kennis hebben van elektronica of zich daarin willen ontwikkelen Learned quality .

Die accuraat, zelfstandig en klantgericht zijn Innate quality .

Als bedrijfsautotechnicus verricht je diagnoses Act , reparatie- en onderhoudswerkzaamheden.

Je zorgt dat de voertuigen bedrijfszeker zijn door een correcte uitvoering van periodiek onderhoud, het correct uitvoeren van alle voorkomende reparaties en het doeltreffend verhelpen van storingen Process .

In de werkplaats voeren we complexe reparaties uit van bussen en stellen we diagnoses(Act evt. met behulp van diagnose- apparatuur en- gereedschap).

We werken met diverse merken waardoor jij je continue blijft ontwikkelen.

Het werk aan elektrische bussen is schoon in tegenstelling tot diesel voertuigen, omdat de werkzaamheden voornamelijk binnenin plaatsvinden dan onder de bus.

Wat breng je mee?

0- 3 jaar relevante werkervaring Learned quality ;.

Bereidheid tot het volgen van(leveranciers) trainingen Act ;.

Een afgeronde opleiding BAT of EBAT Learned quality ;.

Rijbewijs B(eventueel C) en D(of bereid deze te halen);.

Kennis van moderne elektronica, pneumatiek en hydraulica is een pre Learned quality ;.

Bereidheid om in een rooster te willen werken.

Wat bieden wij?

Diplomatoeslag en jouw eigen gereedschapskist.

Vrijheid en zelfstandigheid in je functie.

Een collegiaal en betrokken team wat voor elkaar klaar staat.

Daarnaast werken wij met diverse grote merken, zoals VDL, BYD, Mercedes, MAN etc. En zijn er diverse mogelijkheden voor verdere ontwikkeling.

Verder bieden wij Een goed salaris conform CAO OV, 8 [UNK] vakantiegeld en een eindejaarsuitkering van 1, 5 [UNK] [UNK] [UNK] 400,- bruto per jaar(dit is exclusief toeslagen, mocht dit van toepassing zijn);.

23 tot 29 vakantiedagen, afhankelijk van leeftijd;.

17 of 26 ATV dagen, afhankelijk van leeftijd;.

Een goede pensioenregeling.

Wie zijn wij?

Transdev Nederland geeft mensen de vrijheid om te reizen.

Als toonaangevend mobiliteitsbedrijf zijn we actief in het hele land onder diverse merknamen zoals Connexxion, Hermes en Witte Kruis.

We leveren innovatieve mobiliteitsoplossingen voor personenvervoer en dragen met onze duurzame voertuigen en initiatieven bij aan de energietransitie in ons land.

Bij alles wat we doen staan onze reizigers en medewerkers centraal.

Transdev Nederland heeft 8. 000 medewerkers en vervoert dagelijks zo' n 500. 000 mensen.

Enthousiast?

Ben je enthousiast geworden om aan de slag te gaan als bedrijfsautotechnicus **Attitude + action** ?

Solliciteer dan snel!

Vergeet niet je CV en motivatiebrief in te sturen **Attitude + action** .