university of groningen / faculty of arts

Language and Dialect identification

How Twitter can be used to create a corpus of a specific (regional) language

Lennart Albert Schepers

Bachelor thesis
Informatiekunde
Lennart Albert Schepers
S2922916
July 5, 2020

ABSTRACT

Language identification, especially when nuanced down to regional language identification, is a rather new area in the field of machine learning. This work focusses on the generation of a corpus of messages submitted to Twitter written in both Frisian and the neighboring regional language of Gronings. With the help of queries that consist of a growing list of selected keywords, an annotated corpus of Gronings and Frisian was compiled. On these corpora, three machine learning models are trained: a Naive Bayes model, a logistic regression model and a support vector classification model. All models achieved high accuracy during development. For the task of distinguishing Gronings from non-Gronings a support vector machine reached a weighted f-measure of 0.93. For the task of distinguishing Frisian from non-Frisian a naive Bayes classifier reached a weighted f-measure of 0.83.

CONTENTS

PREFACE

Special thanks to Martijn Bartelds for supervising and guiding the process of writing this thesis. The isolation due to the Corona epidemic made concentrating on tasks like these much harder, but you provided thorough and honest feedback which helped me a lot.

1 INTRODUCTION

As Social media services have exponentially grown in user bases, conversations happen increasingly often through social media (Blommaert, 2012). With over 320 million users worldwide, Twitter is no exception to this [1]. With the growth of text-based social media services like Twitter comes an increased amount of user-generated text that is available online. With its millions of users, Twitter can be a unique source of descriptive linguistic data from all around the world. Providing user generated messages not only in the 34 languages that are supported for automatic labelling, but also in many more regional languages and dialects.

According to Ljubesic et al. (2007) it is impossible to extract information out of a text without knowledge of the language it is written in. Since regional languages and dialects remain mostly unlabeled by Twitter, there exists big untapped potential to retrieve information from such texts by first identifying its language.

To be able to identify the language a text is written in, the text first has to be represented as features. This can be done in multiple ways. Two example are n-grams and the bag-of-words method. After the text is represented as features, the text can be classified. For this task, a machine learning model can be trained on a body of text. This model can then be tested on its accuracy.

A lot of research has been done on the identification on languages this way, but less so on automatic dialect identification. The process of automatically identifying a regional dialect within a known language is harder than identifying a language, because regional languages have less differences with a known language than other languages do (Hoppenbrouwers and Hoppenbrouwers, 2001). Dialect identification can be useful to further understand the regional origin of a speaker, or to detect when 'code switching' occurs (Biadsy et al., 2009). The combination of dialect identification and language identification is therefore a useful task. This why this research focuses on the creation of two corpora from Twitter messages: one of the Frisian language and one of the Gronings dialect.

While the official language of the Netherlands is Dutch, the northern province of Friesland has the second official language of Frisian (Hoppenbrouwers and Hoppenbrouwers, 2001). The Frisian language belongs to the west Germanic branch of the Germanic language family and is closely related to English (Hoppenbrouwers and Hoppenbrouwers, 2001). Frisian is spoken in the Dutch province of Friesland and in the German states of 'Nedersaksen' and 'Sleeswijk-Holstein' (Hoppenbrouwers and Hoppenbrouwers, 2001). The variety of Frisian that is spoken in the Dutch province of Friesland is officially named West Frisian or 'Westerlauwers Fries'. Since this is the variety of Frisian that this research focuses on, will be just referred to as Frisian from now on.

Research in 2004 concluded that around 440.000 people in Friesland speak Fries, with around 350.000 native speakers (Lewis, 2009). A type of barbarism exists for Frisian, which in Dutch is appointed with the term 'frisisme'. A 'frisisme' happens when a word or syntactical structure is formed like it would be in Fries, in another language. An example of this would be the sentence: 'Ik lig op dit stuit in bad'. In proper Dutch, this sentence would be: 'Op dit moment lig ik in bad' and in proper Fries: 'Op dit stuit lig ik yn bad'. In this example, the 'frisisme' happens because the Fries word 'stuit' is mixed with a more Dutch syntactical structure.

[1] https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

Gronings is a collection of multiple varieties of regional languages spoken mainly in the northern Dutch province of Groningen. The usage of Groningen is not just confined to the province of Groningen, as some disagreement exists whether or not the dialect spoken in the northern part of neighboring province Drenthe (referred to as 'Noordervelds' can be allocated to Gronings because of similar phonological characteristics; such as the 'ou' and 'ei' sounds which are typical for a Gronings dialect, rather than 'oe' and 'ie' which would be usual for the 'Drents' dialect (Hoppenbrouwers and Hoppenbrouwers, 2001). Then there is the border with the province of Friesland, where the Gronings variety of 'Westerkwartiers' crosses territory.

The increased importance of Dutch over the last decades in Groningen has influences a hybrid of Dutch and Gronings (Blommaert, 2012). This so called 'Gronings Dutch' is a mixture of Dutch with syntactical or lexical characteristics of Gronings. Examples would be the sentence 'Mag ik een puutje erbij?'. The proper Dutch sentence would be: 'Mag ik een tasje erbij?'. The Dutch Gronings sentence is identical except for the Gronings word: 'puutje'. Another example would be the Dutch Gronings sentence 'je doet me daar geen knup om toe'. In proper Dutch this sentence would be something like 'Je doet er geen knoop om'. Here more differences are visible; the Dutch Gronings sentence contains the Gronings word 'knup' and the Gronings syntactical structure of the place of the words: 'me' and 'ergens om toe'.

Because of the described occurance of so-called 'frisismes' and texts that are hybrids of Dutch and Gronings, corpora will not exclude texts that contain Dutch lexical or syntactical features alongside Gronings or Frisian. ke In this paper, the main question that is trying to be answered is: *'is it possible to create a high quality corpus of a dialect and minority language on Twitter and to then use this corpus to identify tweets in Gronings and Frisian from Dutch tweets with high accuracy?'* This main question is attempted to be answered by the following sub questions:

1. To which extent is it possible to create a high-quality corpus from Tweets that contain the Gronings dialect using seeding terms and geolocation?

2. To which extent is it possible to create a high-quality corpus from Tweets that contain the Frisian language using seeding terms and geolocation?

3. To which extent is it possible with the created corpora to correctly identify Tweets in the Gronings dialect or Frisian language?

2 BACKGROUND

Other than working on an approach of creating corpora, this research will focus on different kinds of approaches at classifying data. One key aspect in this regard is the way that data is represented in features. Scott (2000) reports that most research uses the 'bag of words' method of representation, in which the data is represented as individual words from tokenized text. According to Scott (2000), the 'bag of words' method can be used together with the removal of stop words, stemming (i.e. reducing words to their morphological root), or lemmatization (i.e. using the dictionary form of a word) to reduce the complexity of features. Previous works of research have also shown the usefulness of the n-gram approach of representing features, where in text subsequent features are split in vectors of n items (Scott, 2000). The n-gram approach is especially useful for categorization problems since this approach works the same regardless what language is being used (Martins and Silva, 2005). Errors seem to be limited to the smaller, fragmented parts of the sentence, which is helpful for classifying language on platforms on the internet, where speech might not be standardized all the time (Martins and Silva, 2005).

A problem that the n-gram approach has however, is that the n-gram database can grow to excessively big proportions with a large training set, which is why often an alternative approach is chosen, such as the 'bag of words' approach, where texts are tokenized into vectors (Scott, 2000).

Joachims (1998) and Martins and Silva (2005) have shown the usefulness of 'support vector machine' ('support vector clustering' or SVC), logistic regression and the 'naive Bayes' algorithms to execute the task of assigning language labels to the text. Joachims (1998) found support vector machines to be currently the best performing methods for learning with text data and automatic parameter tuning through grid-search.

2.1 DIALECT IDENTIFICATION

Since not only the Frisian and the Dutch languages will be identified but also the Gronings dialect, this research will also include the field of dialect identification. Dialect identification is the process of automatically determining if text contains content from a dialect and is very similar to the task of language identification. Dialect identification is harder because of the aspect that the subjects of classification are a group of most often close languages, rather than separate languages (Zaidan and Callison-Burch, 2013). Since dialects and corresponding languages often share the same script, large parts of their vocabulary and parts of their syntactical structure, they are harder to classify correctly (Zaidan and Callison-Burch, 2013).

Many research has been done in the last few years on identification of languages, but less research has been done on dialect identification. Most research regarding dialect recognition has been done on Arabic dialectsElgabou and Kazakov (2017) (Mubarak and Darwish, 2014). These works of research focus on creating a multi-dialectal corpus of Arabic through the use of geographical areas on Twitter, which seemed to be highly successful. Elgabou and Kazakov (2017) is relevant to this research because both researches focus on the creation of a corpus and identification of one or more dialects. A difference with Arabic dialects however is that they differ so much from each other sometimes that they might not even be seen as dialects anymore. An example of this is the difference between the 'Maghreb' dialect, spoken in Morocco, and the 'Gulf' dialect, spoken in the UAE (Elgabou and Kazakov,

2017).
Ciobanu et al. (2018) also researched identification of a dialects, in this case dialects of german. Transcripts of spoken dialects in German are used to train SVM classifiers. These classifiers reached an F-score of 62.03 percent. Ciobanu et al. (2018) is relevant to this research because this research will also make use of an SVM classifier to compare results with.
sectionIdentification of Twitter messages

According to Bergsma et al. (2012), correctly classifying a language from messages from Twitter is an especially hard task because of the nature of Twitter messages; they are short, informal of nature and they can be of numerous different languages. Bergsma et al. (2012) gathered their data using the Twitter API and geographical location, which retrieved messages originating from different languages. Languages such as Nepali, Urdu and Ukrainian were then manually annotated. The dataset was then used to train a logistic regression classification model and a partial matching algorithm. The logistic regression model achieved an accuracy score of over 90 percent (Bergsma et al., 2012).

Tratz (2014) focused on classifying Twitter messages of three languages using multinomial logistic regression and SVM's. The research implemented the data set from Bergsma et al. (2012) together with their own that was comprised of Arabic, Farsi and Urdu and got similar results, nearing results of almost 100 percent. The dataset was comprised of 1100 Twitter messages. Then this same model was trained using the Zaidan and Callison-Burch (2011) data set, with 3000 tweets from one Arabic dialect and 3000 tweets from other Arabic dialects, using SVM's. This time the algorithm scored significantly worse, with an accuracy of about 80 percent. Tratz (2014) suggested that the worse score was due to complicated differences between Arabic dialects.

Lastly, research by Artur Kulmizev and Wieling showed a linear support vector machine trained on character features, reaching f-measures of 87.56. Following the research by Artur Kulmizev and Wieling the decision was made that data tokenization could not happen on just a word level but also a character level, which is why this extra comparison is added to the research.

3 DATA AND MATERIAL

3.1 COLLECTION

In this study, the free version of the Twitter API is used. This free version, however, imposes some limitations on the data collection. First there is a maximum for querying Tweets of 180 requests per 15 minutes and Tweets can only be queried that are submitted in the last seven days. In order to build Gronings and Frisian corpora that are large enough for research purposes, a query script was run every week since the beginning of April.

3.1.1 Corpus creation

The Twitter API can be used to restrict results to a certain geolocation-bound area. This area is defined as a certain location, with a value for the radius of the area. This functionality is useful in the case of this research, since restricting results to the respective provinces where Gronings or Frisian are mainly spoken can improve the frequency of tweets in the relevant language of dialect.
The geolocation functionality only supports a radial specified area, which by itself already poses a challenge since the province of Groningen cannot be well specified within a circle. To solve this issue, a general approach was chosen, by choosing the village of 'Ten Boer' with a radius of 24km the largest parted of Groningen's inhabited locations could be reached by the algorithm. Since the shape of the province of Friesland lends itself better to a radial shape, less sacrifices in terms of reach had to be endured with the village of Grou and a radius of 30km covering most of the province. The choice for a single circle for each province was due to the fact that the query script was run locally and already required high computing power. This choice is further discussed in the Discussions section.

Figure 1: A map with the radius of geographical constraints for the Friesland and Groningen

Initially, gathering Twitter messages will be solely done by geolocation, without any terms as a query. For both Gronings and Frisian, the first 180 messages of that are the result of this process will be manually annotated by a native speaker. The number 180 means that these messages can be gathered without invoking a delay due to the limits of the API. Of all messages that are labeled as Gronings or Frisian, keyword sets are developed to increase the accuracy of finding relevant Tweets. These keyword sets are made by tokenizing and then creating two frequency lists of the tokens of all messages that were labeled as Gronings and Fries. For N messages that were manually labeled as one of these categories, N keywords were chosen. The choice of extracting the same amount of keywords as the number of documents comes from the fact that each document has at least one word in the target (regional) language. The resulting keywords are also annotated by the same native speaker to be of the appropriate category. This process is continued until 50 keywords are found. This number was chosen because 50 seemed adequate enough to continue to the next step of retrieving more keywords, whereas lower numbers provided too little results.

The next step, in order to extend the number of keywords, is calculating terms that co-occur with keywords on a higher-than-chance basis. This is done by using pointwise Mutual Information. PMI quantifies the chance of two words co-occurring with each other, while taking the frequency of single words into account. As can be seen in equation 1, PMI is the logarithmic probability of the co-occurrence of words a and b, divided by the independent probability of words a and b. A positive value for PMI signifies that words co-occur more frequently than would be expected when independence is assumed. Terms that have a positive PMI score for co-locations with existing keywords will be manually annotated to make sure they belong to Gronings or Frisian. These new keywords are then added to the existing list if this is the case. This process is repeated after each week, after new Twitter messages are gathered. The list of keywords can be found in the online repository [1]

$$\mathrm{PMI}(a,b) = \log\left(\frac{P(a,b)}{P(\mathrm{a})\mathrm{P}(\mathrm{b})}\right) \tag{1}$$

[1] https://github.com/lennartschepers/scriptie

The API will be queried weekly for at least 2 months searching for messages that contain at least one of the keywords, and have a geographic location tag that is within the limitations that are specified for Gronings and Frisian. Duplicates will be filtered out of the process, so every message is unique. This prevents any over fitting on spam that might occur.

3.2 ANNOTATION

The first section that will be annotated are the keywords. 50 possible keywords that are chosen by frequency are annotated to make sure that they are a Gronings or Frisian term. For Gronings, the native speaker from Groningen will annotate every keyword as either Gronings or not-Gronings and for Frisian vice versa. Keywords found through high PMI scores are continuously reviewed by the annotators in the same way on a weekly basis.

Naturally, not only the keywords are manually tagged, but also the Twitter messages that are gathered as well. The goal of annotating the data set of Twitter messages is to make a data set with predefined categories, so a machine learning algorithm can be trained and tested in a proper way, without making false assumptions. Without annotated data, supervised machine learning will be impossible.
For Frisian, Twitter messages are tagged as 'FRI' when containing Frisian language and as 'NO' when no Frisian language seems to be present in the message. For Gronings these labels were 'GRO' and 'NO'. 1000 Gronings tweets were manually identified by native speaker from Groningen and 1000 Frisian tweets were manually identified by a native speaker from Friesland.

The native speakers for both (regional) languages were given the following annotation guidelines: A message should be tagged as 'FRI'/'GRO' when the native speaker recognizes lexicon that belong to Gronings/Frisian. When at least one word is recognized as such, the message should be labeled with 'FRI'/'GRO. The rest of the message can be in Dutch. If a word is recognized as Frisian/Gronings but the rest of the Tweet does not seem to be in Dutch or that specific (regional) language, the message can not be tagged as Frisian/Gronings. This is because a multitude of languages are found in both provinces, even with the selected keyword list. Annotating a different language based on (false) cognates obviously does not help the machine learning models to classify the correct dialect or language. Since the native language of Friesland and Groningen is Dutch, we can safely assume that 'code-switching' takes place between their regional language and Dutch. This is why the annotator can only assume a 'FRI'/'GRO' label when at least one word of their (regional) language is recognized, with the rest being either Dutch or that same language. The word that is found can also be recognized, not as a Frisian/-Gronings word, but as a Dutch word, that is changed to have clear lexical features of Frisian/Gronings. This is also sufficient for a 'FRI'/'GRO' label. This research is about messages that *contain* a specific regional language, which is why one element is already deemed sufficient.

For each Fries and Gronings, the manual annotation part is done by one person. This begs the explanation of a possible annotation bias. Since only one native speaker annotated the foundation of this research, there is a possibility of inaccuracies when the annotator labels very selective documents. A classifier might for example perform unrealistically well, when for Frisian mostly documents are annotated that contain specific elements, not necessarily related to Fries, which a classifier can easily identify. One annotator might select a group of documents that

would not necessarily reflect a random group of documents of a particular language. All annotated data can be found in the online repository [2]

3.3 PRE-PROCESSING

As far as pre-processing goes, some elements of Twitter messages will be removed. Usernames and hashtags will discarded, since these are characteristic of Twitter messages in general and not for Gronings or Frisian in particular. The goal of this research is to use the textual information of the messages itself, and not the information that specific to Twitter. The research focuses on natural language processing, rather than the identification of prevalence that hashtags or usernames might have. Not removing these features could lead to over fitting, since recent trends could influence the prevalence of certain hashtags and only Twitter messages of the last two months are used.
Hyperlinks, 'RT' and emojis will be removed for the same reason; they are not the point of this research. All punctuation will be removed equally and text will be turned to lowercase.

3.4 DATA DISTRIBUTION

The result of pre-processing is a data set that can be used by classification models. The data set will be split into a train, development and test set. The training set is used to generalize the classification model to predict unseen data. The purpose of training data is to find the optimal parameters of classification models and for the models to learn about the patterns that the training set might have.
The Development set is the part of the data that is used to rank all models that are used in terms of how accurate they are in their predictions the classification model to predict unseen data. The purpose of training data is to find the optimal parameters of classification models and for the models to learn about the patterns that the training set might have.
In this research, the Development set is the part of the data that is used to rank classification models' performance through different kinds of feature selection. To prevent any bias from occurring, it is important that the final test set is held out from earlier classification where parameters and feature representation are still being experimented with. This is why the development set exists to compare the effect of different kinds of feature representation. The representation with the best score for each model is used on the final Test set. In this final set, different classifiers will be compared to each other.
Following industry standards, the train set will be comprised of 80 percent of the data set, the development and test sets will both be comprised of 10 percent of the data set.
The full data set for Frisian consists of 1000 messages that contain Frisian, and about 500 messages that do not contain Frisian. This ratio of two to one is the same ratio that was found on the Twitter API when querying for messages that contained at least one of the most recent list of keywords within the geographically limited area. This means that for Frisian, the train set will consist of 800 Frisian Tweets and 400 non-Frisian Tweets. The development and test sets then consist of 100 Frisian tweets and 50 non-Frisian Tweets.
Gronings Tweets seemed much less prevalent. This is why a higher priority was chosen for messages that contained at least two keywords. These messages became candidate for annotation before messages that contained just one keyword. After annotating with this heuristic in mind, a ratio of about 1/1.5, Gronings to non-

[2] https://github.com/lennartschepers/scriptie

Gronings Tweets were annotated with the last version of the keyword set. This is why the data set for Gronings has 1000 Tweets that contains Gronings and 1500 Tweets that do not contain Gronings. This means that, for Gronings, the training set consists of 800 Gronings Tweets, 1200 non-Gronings Tweets. The development and test sets are comprised of 100 Gronings and 150 non-Gronings Tweets.

4 METHODOLOGY

4.1 FEATURES

In order to use the data to train a classifier, the data set will have to be represented as features. This research will use different methods of representation and compare their results. These features are computed by tokenization, which is the process of dividing a text into meaningful subsections, or tokens. In this research, tokenization is done by diving sentences into words that are limited by white space. In addition to this, tokenization will also happen by single characters.
First,the 'bag of words' method will be used. The *bag* in this case is a multi set of the tokens in a Twitter message. A multi set is a data structure that, like a set, disregards the original order of the words. Unlike a set though, multiplicity of tokens is taken into account; the same word can occur more than once in a 'bag of words'. The count of each word in a document, or Twitter message, is registered. In this way, a Twitter message is converted from text, to a vector of numbers, where every number is the count for each word in the set of total words. This often comes down to a set of mostly zeros. The process of converting a document to this kind of representation is called 'vectorization'.

Another method of representation that this research will use is the 'bag of n-grams' approach. While a 'bag of words' representation does not keep track of word order, a 'bag of n-gram' representation does. An 'n-gram' represents a text as adjacent sequences of n words or tokens, where n is the amount of words or tokens that each sequence has. A bag of words can similarly be seen as a bag of unigrams. Other than the 'bag of unigrams', this research will also make use of bi-grams (2-grams), trigrams (3-grams), 4-grams and 5-grams and compare the results.

4.2 TF-IDF

The previously mentioned methods of vectorization produce vectors of the count of words or n-grams. In addition this form of vectorization, another form called 'TF-IDF' will be computed. TF-IDF stands for 'term frequency-inverse document frequency. This measure not only takes into account how often a word or token occurs in a document, but also how (in)frequent a token across all documents. This means that word that are frequent in documents in general will provide a lower value in the resulting vector of a document. This method provides a way to associate tokens in documents with how relevant they are to that document. A machine learning model might classify differently with this extra information. This is why TF-IDF is used, to see if it has positive effects on the classifiers.

4.3 EXECUTION

The purpose of this research is to create two corpora and to see to which extend these corpora can be used to automatically identify tweets that contain Gronings or Frisian language. To be able to automatically create a corpus of Fries or Gronings, a model needs to be trained to accurately identify Gronings and Fries documents

from Dutch documents. The two different models that will be used for the classification task are the Naive Bayes model, the support vector machine model and the logistic regression model.

4.3.1 Naive Bayes

The Naive Bayes model is a type of algorithm that makes use of Bayes' Theorem and probability theory to predict the label of a text. Bayes Theorem is an equation that can be used to calculate the probability of a hypothesis being true, in relation with the probability of the hypothesis not being true. In the case of this research, Bayes' Theorem can be used to calculate the probability that a certain Twitter message is written in (regional) language X, in relation with the probability of a certain Twitter message is *not* written in (regional) language X (see equation 2, where D is a document and K is a category). Naive Bayes makes use of probability theory by calculating for each text the probability for every label and then choosing the highest probability. The term "Naive" in Naive Bayes refers to the assumption in the model that features are independent, rather than taking certain distributions into account.

$$P(D \mid K) = \frac{P(K \mid D)P(D)}{P(K)} \tag{2}$$

4.3.2 Support Vector Machine

The support vector machine, or SVM, is a classifier that can learn to distinguish different categories, or classes, by representing the data points in a particular space and then by choosing a 'hyperplane' that most adequately divides data points of different categories. A hyperplane is a subspace that is one dimension lower than the corresponding space it is in. If the data is in a two-dimensional space (e.g. as dots on a graph) then the hyperplane that the SVM will attempt to separate the data with will be one-dimensional (e.g. a line). The hyperplane that offers the most separation (the biggest margin) between the different categories, will be chosen to classify further data with. The way the SVM calculates the hyperplane that provides the biggest margin between data of different categories is by searching for data points of different categories, which are named 'support vectors'. This explains the name of the model; the closest points of different categories form a vector and the best line for separation of categories is 'supported' by the closest data points.

Support vector machines have a way to automatically calculate the best performing parameters, called 'grid-search'. This process works by giving a possible range of options for each parameter that the SVM can work with in the train set, after which the best performing parameters are chosen.

4.3.3 Logistic Regression

The last classification model that is used in this research is logistic regression. Logistic regression is a regression model that predicts the chance that a data point belongs to category. To do this, logistic regression models the data using a 'sigmoid' function (see equation 3). The 'e' in equation 3 stands for 'Eulers' number, which roughly comes down to 2.7183. Since this research focuses on a model predicting if a message belongs to a certain category or not, binomial logistic regression will be used, which means that the output can be of only two types. The logistic regres-

sion model is heavily depended on a threshold. The threshold that provides best separation of data points will be used.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

4.4 EVALUATION

The Naive Bayes,SVM and logistic regression models will classify the documents in this training for each of the feature representation types. The result of the classification will be compared to the golden standard of correct labels. To evaluate the classification of the different models in conjunction with the different feature representation types, different measures are calculated. These measures are based on:

1. 'True positive' (a model correctly classified a document *positively*, or as 'having a certain label)
2. 'True negative' (a model correctly classifies a document *negatively*, or as 'not having a certain label)
3. 'False positive (a model incorrectly classifies a document positively)
4. 'False negative (a model incorrectly classifies a document negatively)

From these values, three different measures will be calculated:

4.4.1 precision

Precision is the proportion of true positives among all positives:
$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$

4.4.2 recall

Recall calculates how many actual positives are classified by the model as true positive:
$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$

4.4.3 F-measure

The f-measure, or F1-score, calculates the weighted average of precision and recall where extreme values are penalized:
$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$ The weighted average of the s-measures of both categories will be the main form of evaluation used in graphs. This means that the f-measure is multiplied with the amount of documents in that category and then averaged out. Other types of evaluation will be used as further explanation.

Labels for the languages are translated to one of three integers for the machine learning algorithms are shown in figures 2 and 3:

FRI	0
NO	1

Figure 2: The original annotated label (Frisian and non-Frisian vs the output that the models will give

GRO	0
NO	1

Figure 3: The original annotated label (Gronings and non-Gronings vs the output that the models will give

5 RESULTS AND DISCUSSION

5.1 VECTORIZATION COMPARISON

As described earlier, for each model and feature representation, results will be compared of count vectorization versus TF-IDF vectorization.

5.1.1 SVM

As we can see in figure 4 and 5, in the Gronings set the SVM resulted in a very slight change of weighted f-measures when comparing count vectorization versus TF-IDF vectorization. When tokens are represented by character, both vectorization types give almost the same f-measure as output. The bag of words representation has the count vectorization at a 0.01 higher weighted f-measure and at trigrams does the TF-IDF vectorization have a 0.01 higher score. When tokenization happens on a word level, the difference is again very marginal; with an output of 0.02 higher for count vectorization at bigram representation and a 0.01 higher weighted f-measure for TF-IDF at 4-gram level. TF-IDF vectorization seems to be performing better, since the absolute highest weighted f-measure is in both comparisons yielded by using TF-IDF.

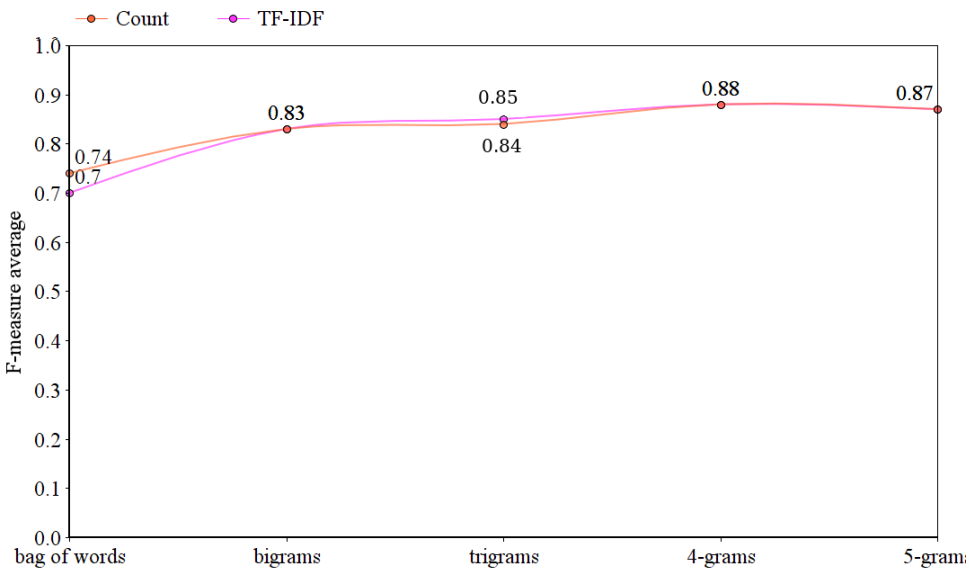

Figure 4: Output of the SVM on the Gronings development set where features are represented as characters

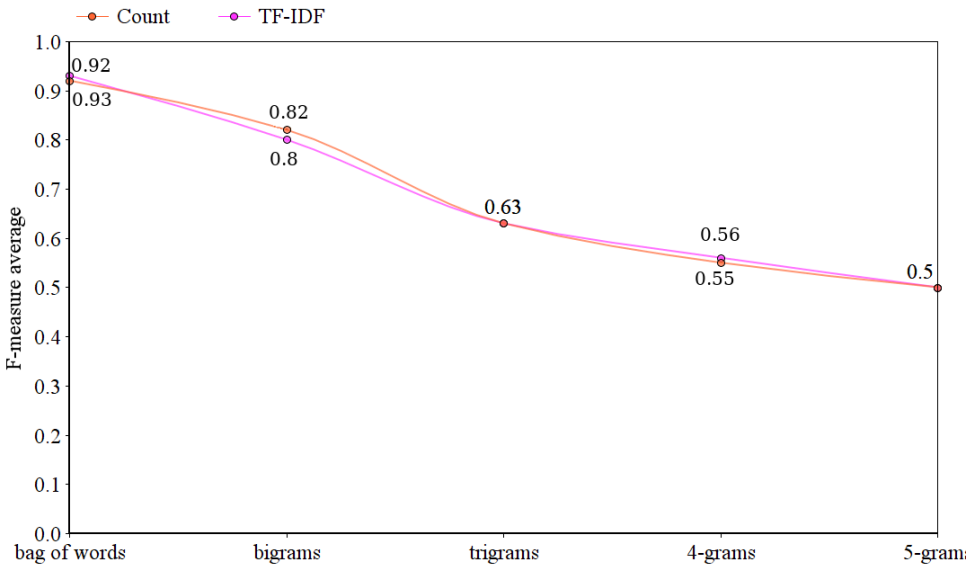

Figure 5: Output of the SVM on the Gronings development set where features are represented as words

In figure 6 we can see that for Frisian, slightly bigger differences occur when text is tokenized by character. Only for a 'bag of words' representation is there a slightly higher weighted f-measure for count vectorization. From bigram to 5-gram, TF-IDF vectorization gives either higher, or equal weighted f-measure output. This is why TF-IDF is the preferred choice in terms of vectorization when text is tokenized on a character level. Figure 7 shows us a smaller difference when tokenization happens on a word-level. Here f-measure outputs are very similar again, with TF-IDF only seeming favorable at 'bag of words'. Count vectorization produces either slightly higher or equal results from bigram representation on. Since TF-IDF again produces the highest f-measure however, it will be chosen as the preferred vectorization method for word tokenization on this set as well.

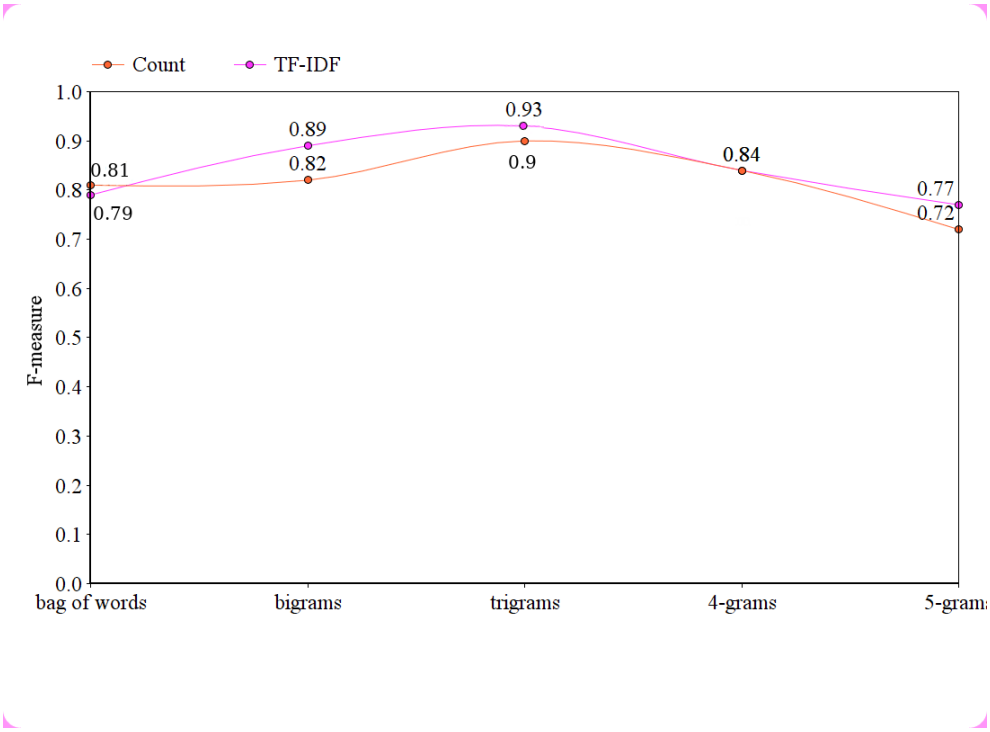

Figure 6: Output of the SVM on the Frisian development set where features are represented as characters

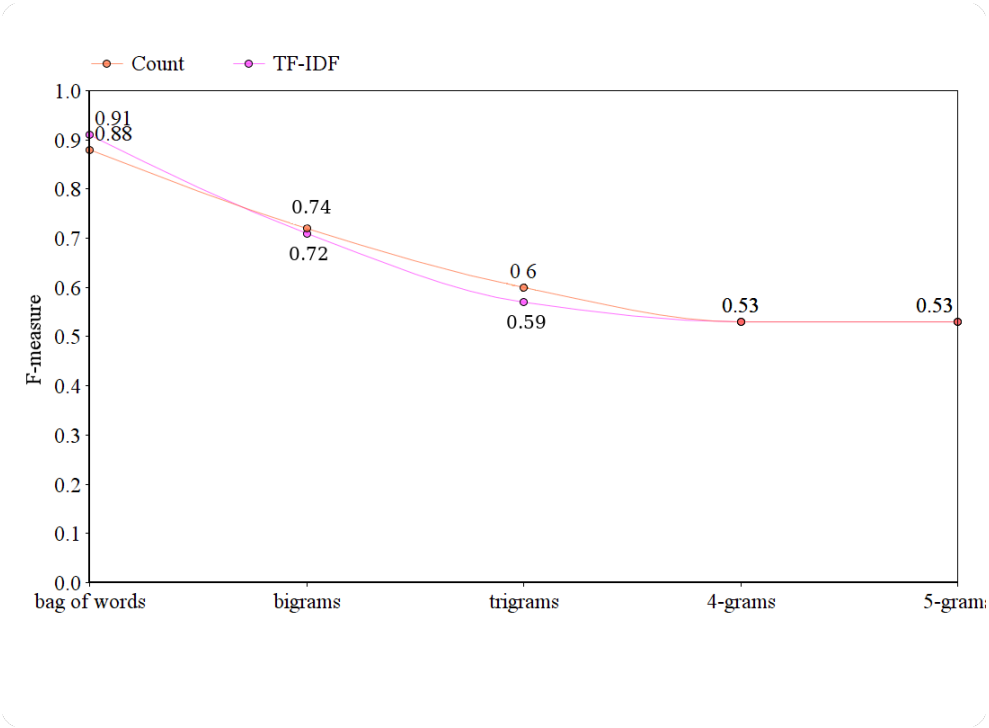

Figure 7: Output of the SVM on the Frisian development set where features are represented as words

5.1.2 Logistic Regression

Figure 8 shows substantial differences between vectorization types when logistic regression is applied on the Gronings development set, tokenized by character. Count

vectorization seems clearly favorable. Figure 9 shows similar results when tokenization happens on a word level, with count vectorization yielding higher measures on all levels of representation.

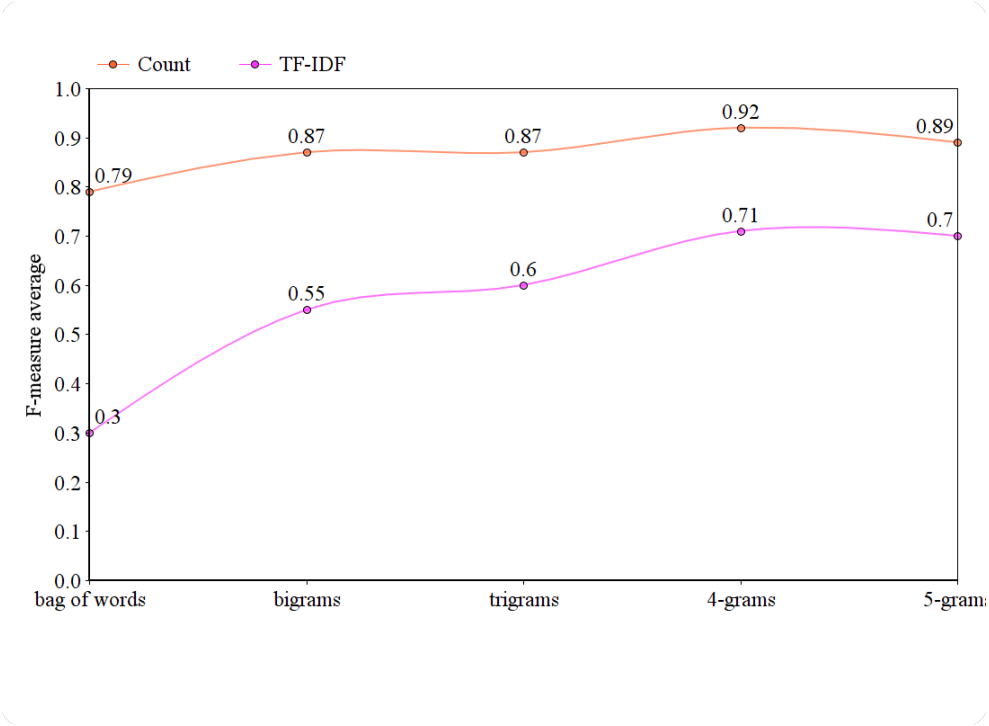

Figure 8: Output of the logistic regression model on the Gronings development set where features are represented as characters

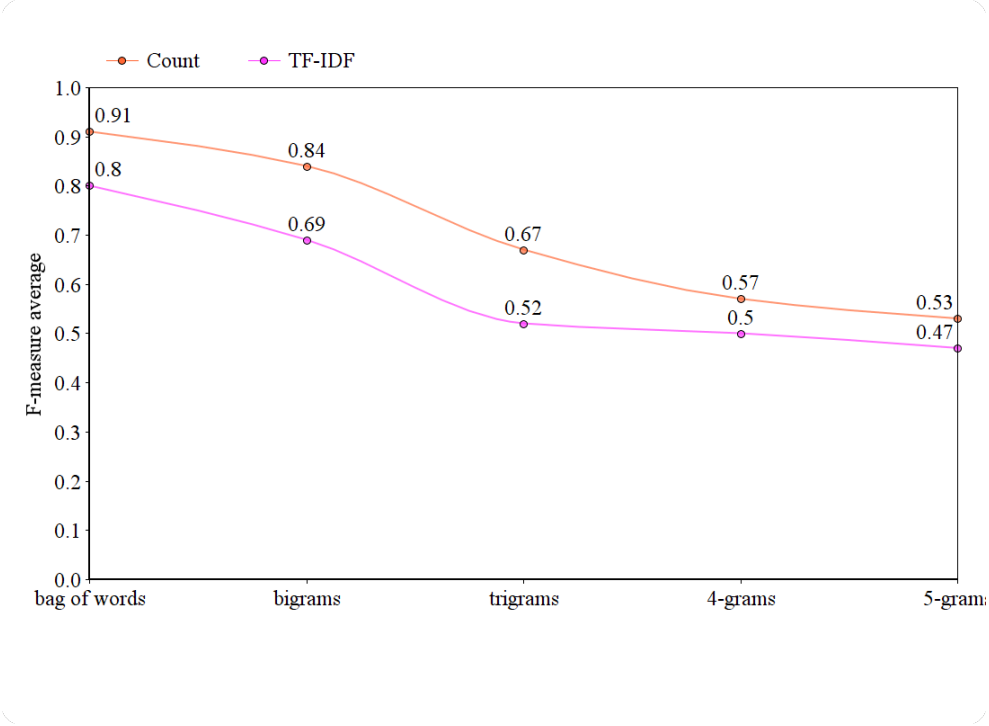

Figure 9: Output of the logistic regression model on the Gronings development set where features are represented as words

We can see this trend continue on Figure 10, where the Frisian development set is represents features as characters. Count vectorization leads to overall better scores. When these features are represented as words however, this difference seems smaller. On figure 11 we can see that count vectorization still yields higher f-measures until trigrams, but then both vectorization types seem to converge to 0.53. For both tokenization types, count vectorization seems to be preferred while using logistic regression.

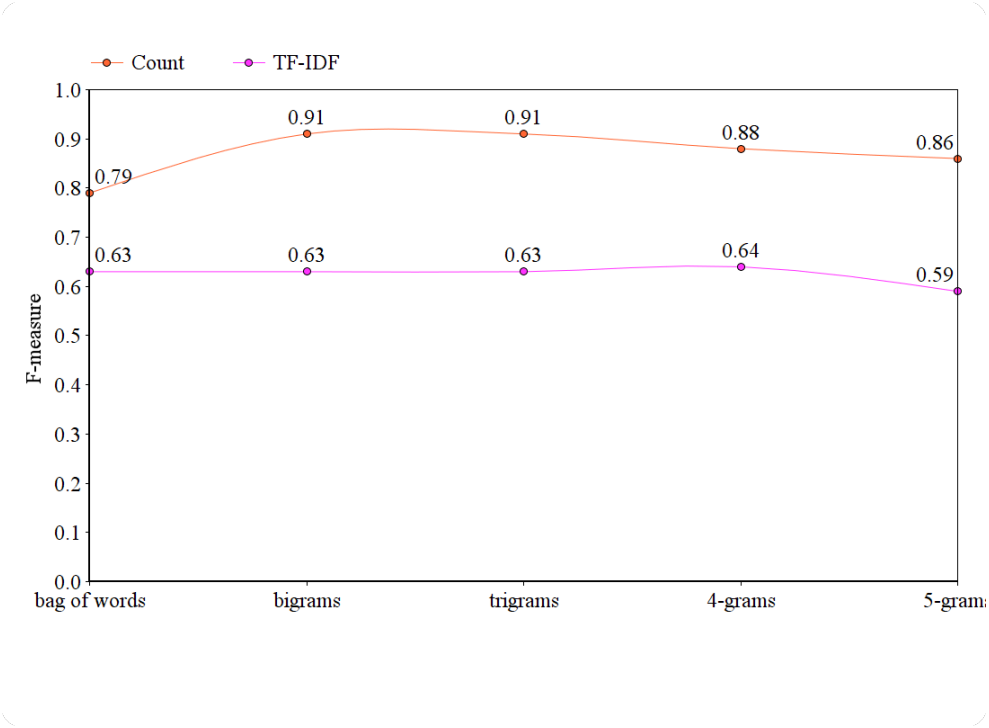

Figure 10: Output of the logistic regression model on the Frisian development set where features are represented as characters

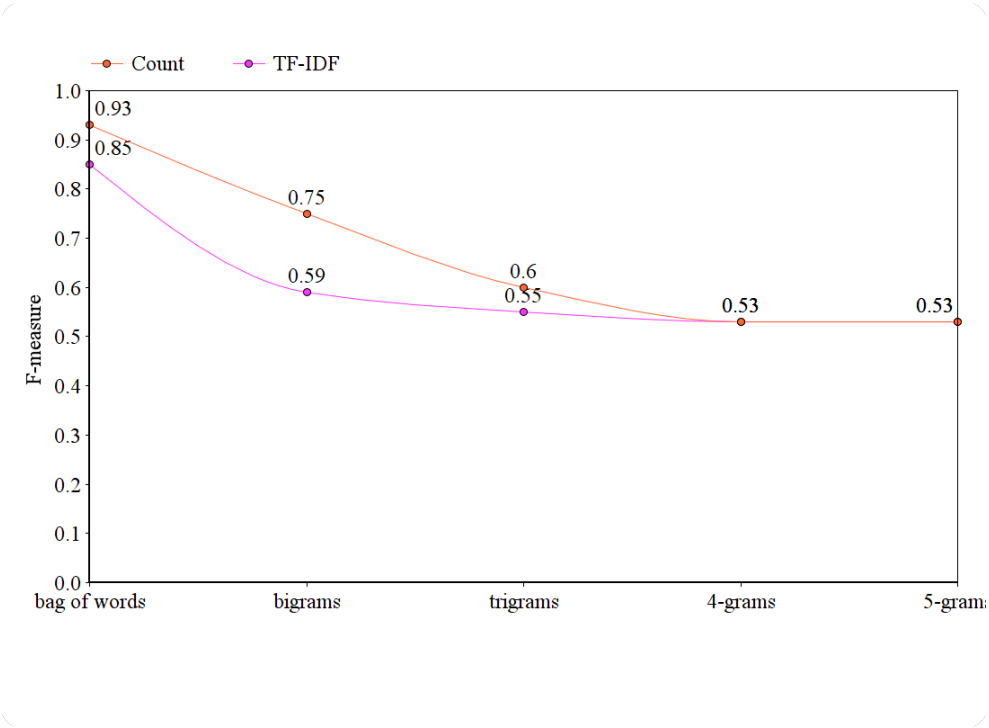

Figure 11: Output of the logistic regression model on the Frisian development set where features are represented as words

5.1.3 Naive Bayes

On the Gronings development set we can see on figure 12 that for character-level tokenization, the highest f-measures are yielded by count vectorization, except for bigram representation. When features are represented by bigrams, TF-IDF just barely yields a higher weighted f-measure by 0.01. This is not the case when tokenization happens by words. As we can see on figure 13, count vectorization yields a stable higher f-measure. In both cases the absolute highest weighted f-measure is produces by using count vectorization

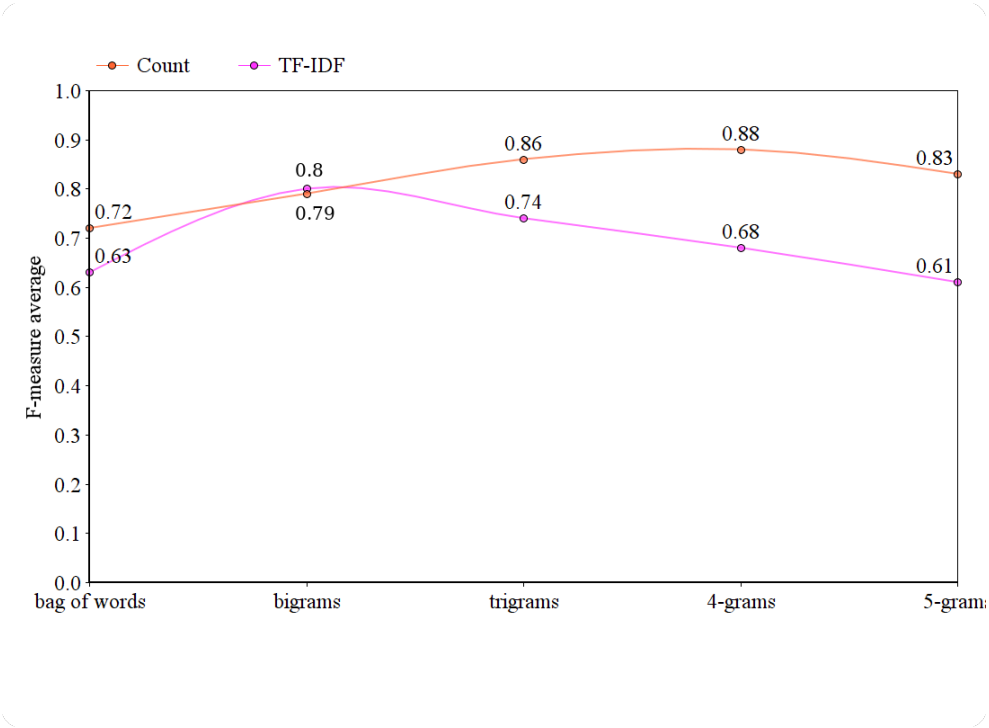

Figure 12: Output of the naive Bayes model on the Gronings development set where features are represented as characters

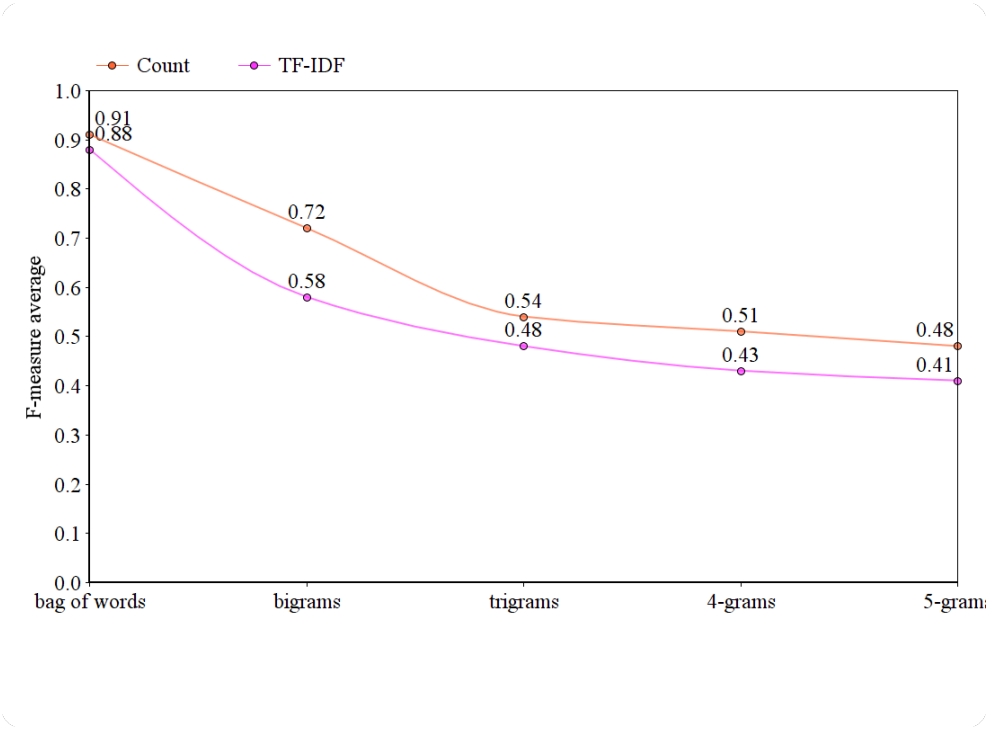

Figure 13: Output of the naive Bayes model on the Gronings development set where features are represented as words

On figure 14 we see a trend that looks slightly similar to figure 12, where TF-IDF vectorization only yields a better result when features are represented as bi-grams. Every other representation type outputs a better weighted f-measure with count vectorization, with the absolute highest score being 0.93 for trigrams based on character-level tokenization with count vectorization. For word-level tokeniza-

tion, displayed in figure 15, we can see a similar trend as the one in figure 11, were count vectorization yields the best f-measure when features are represented as 'bag of words', but again in figure 15 we see that from trigram feature representation on, f-measures converge to 0.53. Count vectorization again seems preferred.

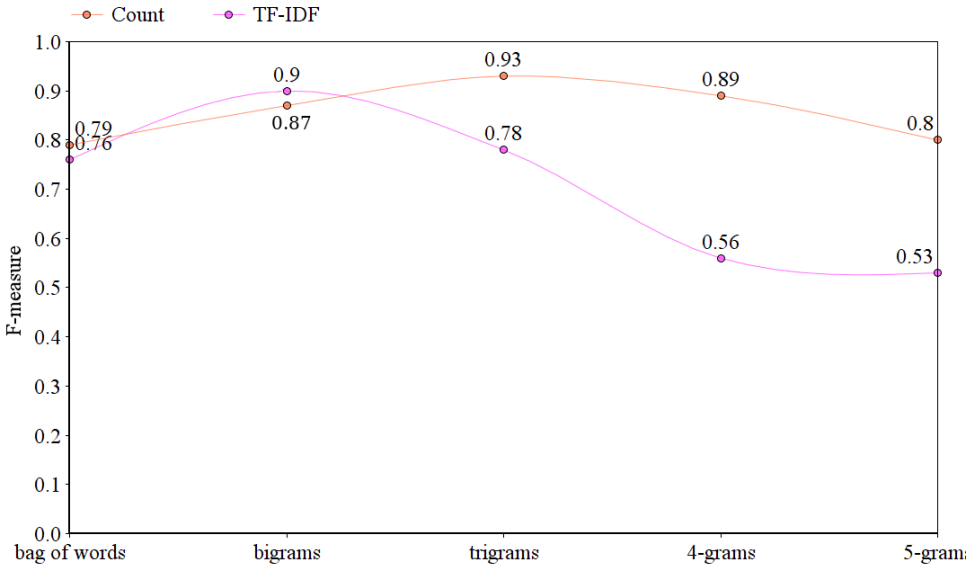

Figure 14: Output of the naive Bayes model on the Frisian development set where features are represented as characters

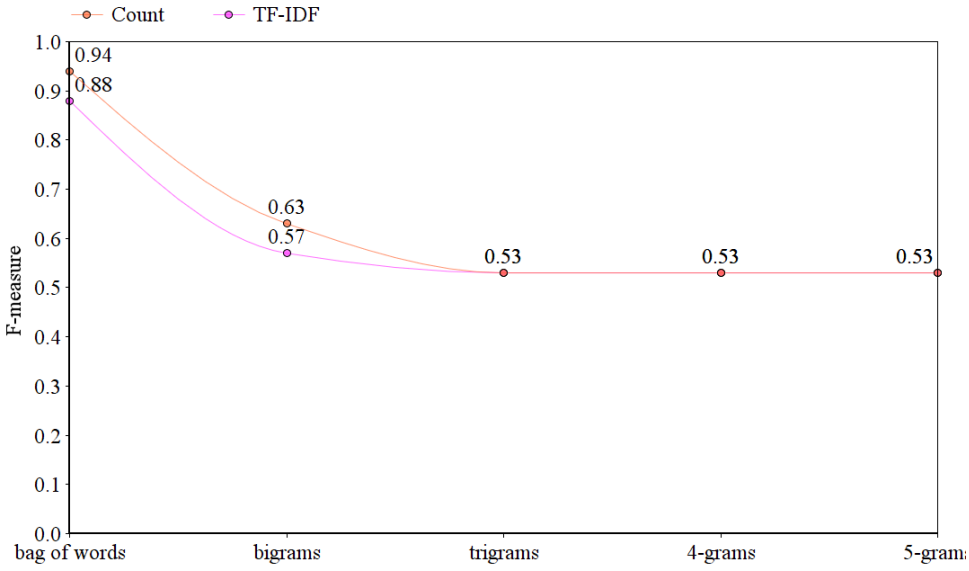

Figure 15: Output of the naive Bayes model on the Frisian development set where features are represented as words

5.2 MODEL COMPARISON

On both Gronings and Frisian development sets, for character and word based tokenization, the classification methods are compared to each other using the vectorization method that yielded the highest result respectively. We can see all three classification methods' performance on the Gronings development set where tokenization happens on a character-level. All models here use count vectorization. As we can see, logistic regression yields the highest results all around on this set, producing a weighted f-measure of 0.92 for 4-grams.

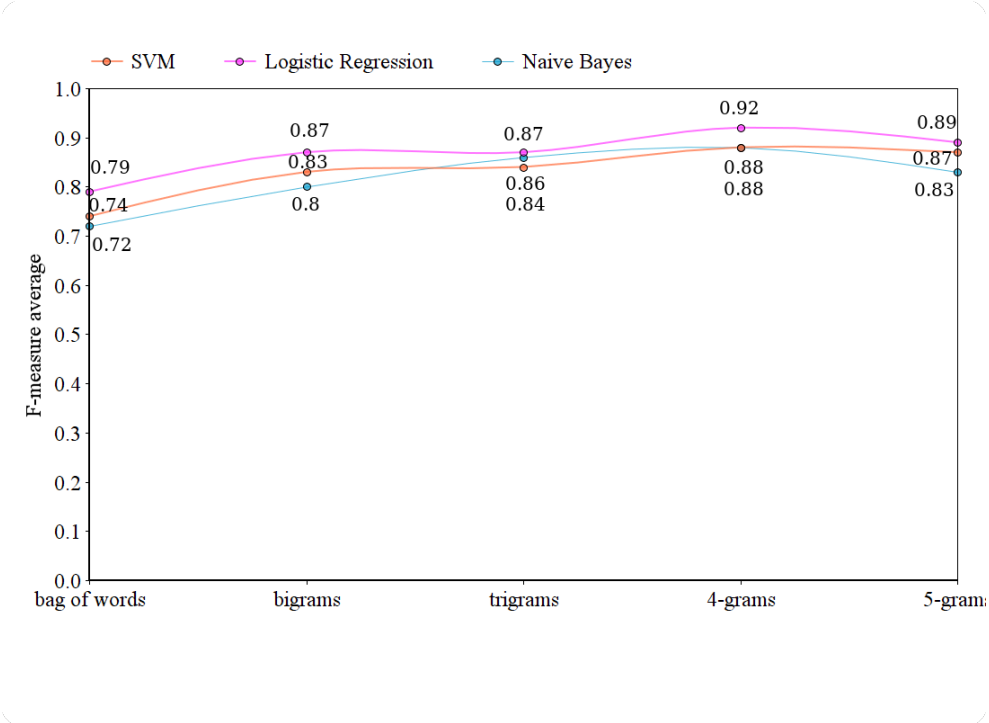

Figure 16: A comparison of all three classification models with their respective chosen vectorization type, on the Gronings development set where tokenization is done on a character level.

When the set is tokenized by words, the results are different. We can see in figure 17 that the highest score is achieved by SVM on bag of word, with a weighted f-measure of 0.93. The highest f-measures for all the other n-grams are for logistic regression. Since the overall performance was best by using an SVM, this classifier will be chosen for further comparison.

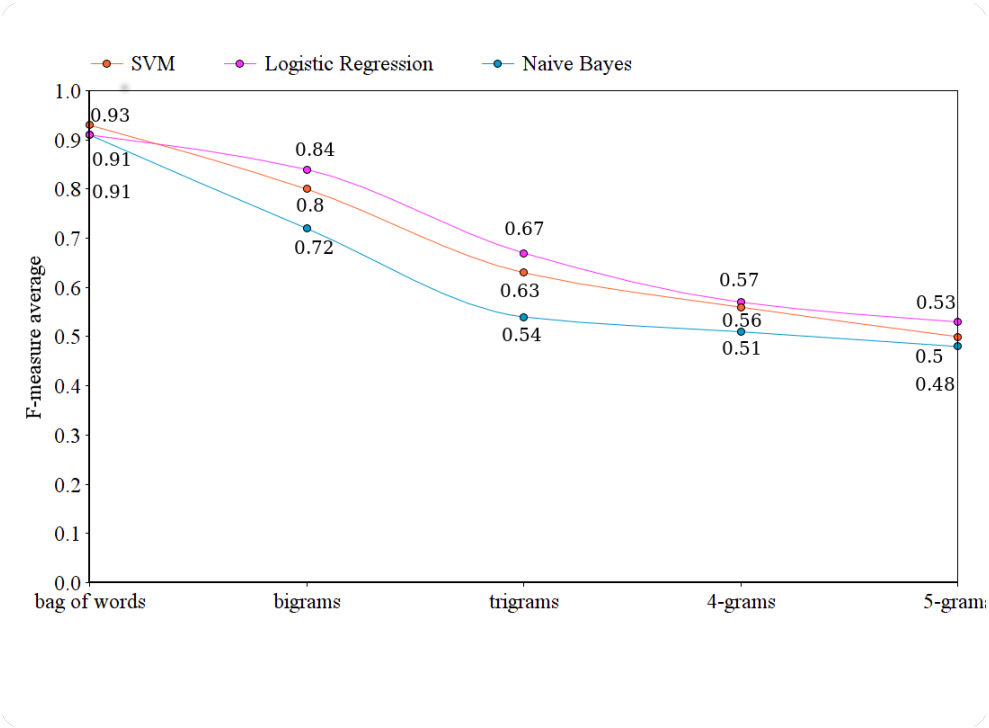

Figure 17: A comparison of all three classification models with their respective chosen vectorization type, on the Gronings development set where tokenization is done on a word level.

Then for the Frisian data set, we compare the models with text that is tokenized as characters. The SVM here uses TF-IDF vectorization and the others use count vectorization. In figure 18 we can see that the three models seem to follow a similar curve, with logistic regression performing the best for 'bag of words' and bigrams. For trigrams however, the SVM and naive Bayes outperform logistic regression, with a weighted f-measure of 0.93 for both models. This is the highest overall f-measure in this comparison. Since Both classifiers are tied in this respect, the classifier with the highest average of weighted f-measures in figure 18 will be chosen. The average of weighted f-measures for SVM in this plot is 0.844 and the average weighted f-measure for naive Bayes is 0.85. This means that naive Bayes will be chosen for further comparison.

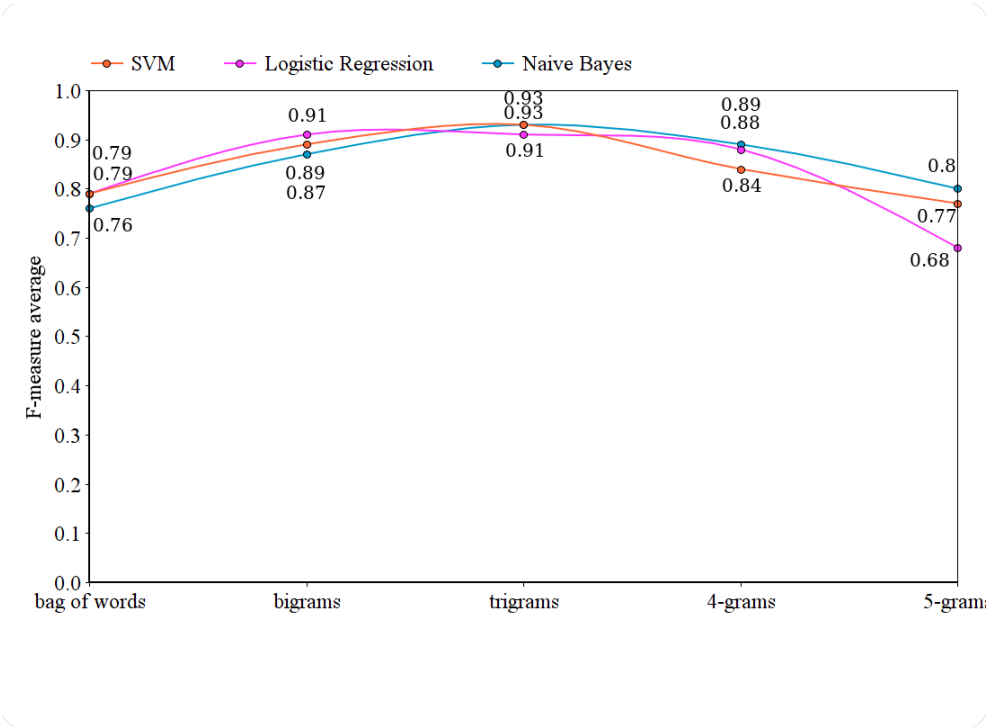

Figure 18: A comparison of all three classification models with their respective chosen vectorization type, on the Frisian development set where tokenization is done on a character level.

In figure 19 we can see the comparison for the Frisian data set when tokenization happens on a word level. Surprisingly enough, the naive Bayes has the best performance on 'bag of words', with a weighted f-measure of 0.94, which is the highest score yet. On further n-grams however, naive Bayes falls short, being either the lowest performing classifier or equal on 4-grams and 5-grams. As it produces the highest weighted f-measure, naive Bayes, with the Frisian development set tokenized by words will be chosen for further comparison.

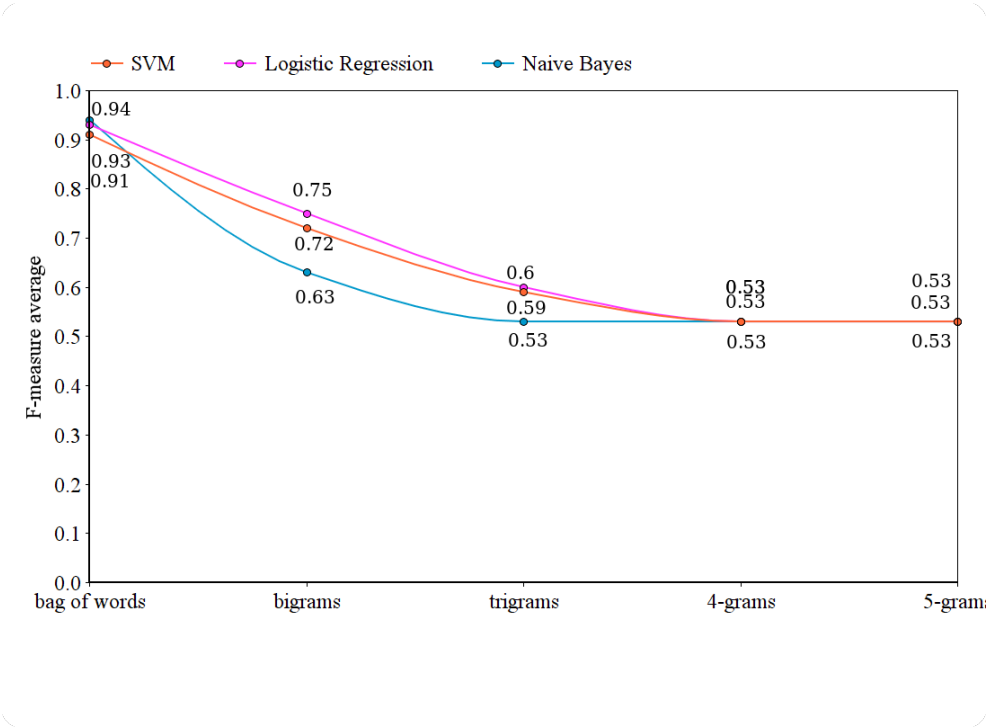

Figure 19: A comparison of all three classification models with their respective chosen vectorization type, on the Frisian development set where tokenization is done on a word level.

5.3 FURTHER COMPARISON

Now that the best performing combinations of vectorization and classifier type are chosen for each language, they will be further compared by tokenization type. These final comparisons will be evaluated using the test set of both Gronings and Frisian. In figure 20 we can see that, for the Gronings test set, word tokenization produces the best weighted f-measure. We can see a trend that for 'bag of words' and bigrams, word tokenization performs better, and for the other n-grams character tokenization performs better. Since word tokenization produces the best f-measure, it will be chosen for language comparison. In the table under figure 20 we can see how the highest f-measures for both character and word level tokenization came about. We can see that the precision for category 1 (non-Gronings) when tokenization happens by character is 0.88. In addition to that, the recall for category 0 (Gronings) is 0.84. Both of these values are low compared to the precision and recall scores for word tokenization. This is why word tokenization scored slightly better in this comparison.

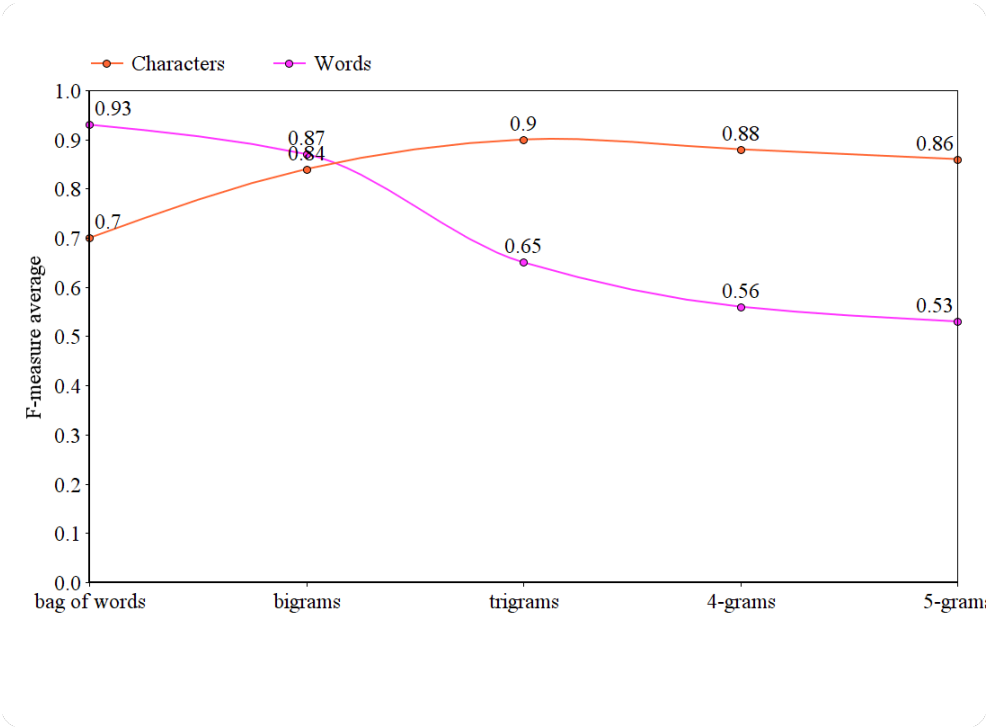

Figure 20: A comparison of the best performing models with their respective chosen vectorization types, when tokenization happens on a character and a word level for the Gronings test set.

tokenization	category	precision	recall	f-measure
character	0	0.93	0.84	0.88
character	1	0.88	0.95	0.91
word	0	0.91	0.92	0.92
word	1	0.94	0.93	0.93

In figure 21 we can see that for Frisian, the tokenization types follow a similar trend as can be seen in figure 20; for 'bag of words', word level tokenization yields the absolute highest score. On all other n-grams however, character level tokenization performs better. Since word level tokenization produces the highest f-measure, it will be chosen for language comparison. In the table under figure 21 we can see how the highest weighted f-measures came about. What becomes evident is that for both forms of tokenization, category 1 (non-Frisian) have both relatively low precision and recall scores. This explains why both f-measures seem to be on the low side. It seems that the classifier for this data set has a bias towards category 0.

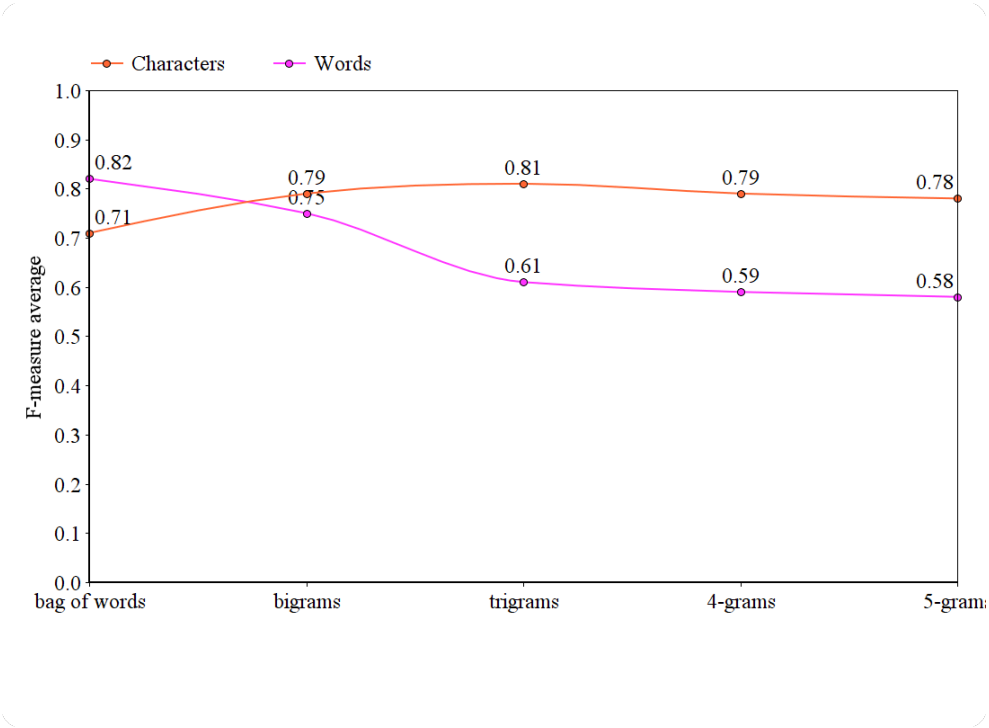

Figure 21: A comparison of the best performing models with their respective chosen vectorization types, when tokenization happens on a character and a word level for the Frisian test set.

tokenization	category	precision	recall	f-measure
character	0	0.87	0.85	0.86
character	1	0.70	0.74	0.72
word	0	0.86	0.89	0.87
word	1	0.75	0.70	0.73

Then finally, the best combination of tokenization and vectorization with the best performing classifier will be used to compare the Gronings and Frisian data sets. For the Gronings test set, SVM was chosen as the preferred classifier, with TF-IDF vectorization and tokenization on a word level. The Frisian test was evaluated using a logistic regression classifier, with count vectorization and also tokenization on a word level. In figure 22 we can see that the Gronings data set scored better than the Frisian data set. The highest weighted f-measure are reported as 0.93 for Gronings and 0.82 for Frisian, both for 'bag of words'. The Gronings set also has better f-measures for bigrams and trigrams. For 4-grams and 5-grams, the Frisian test set seems to yield better f-measures. Overall we can conclude that the Gronings data set produces better f-measures than the Frisian data set. This could signify that, at least in the test set, the Gronings data set is more distinctive from the non-Gronings set, than Frisian is with its non-Frisian counterpart.

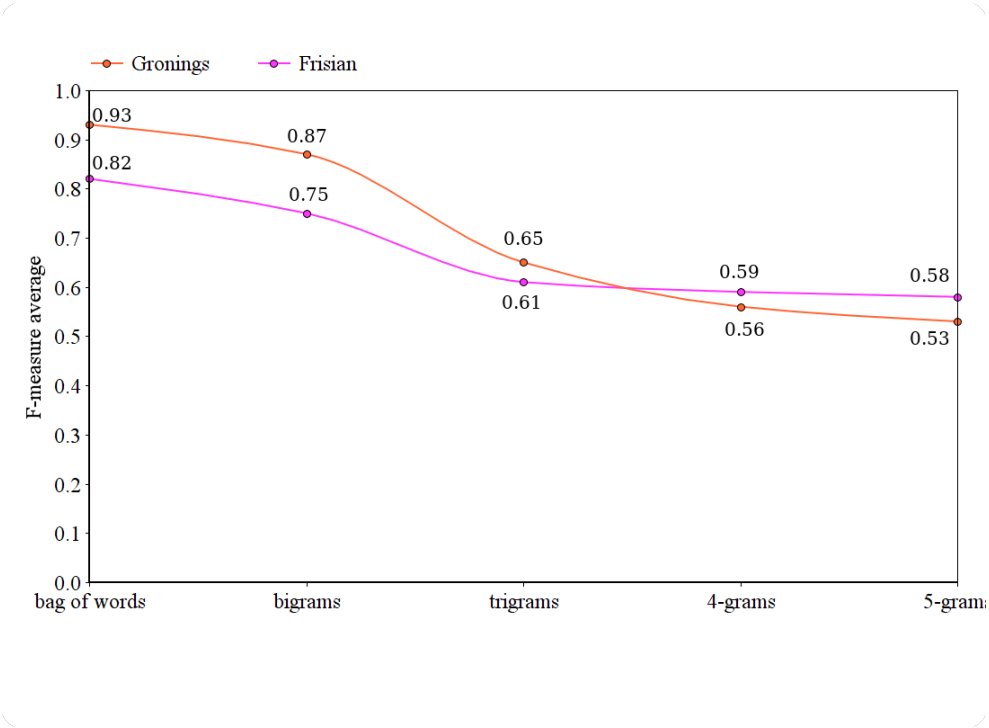

Figure 22: A comparison of the Gronings test set (SVM) and the Frisian test set (naive Bayes)

5.4 DISCUSSION

In figure 22 we can see that the f-measures for Frisian are unexpectedly low. Since the same parameters on the development yielded an f-measure of 0.94, more than a top value of 0.82 was expected. In the table under figure 21, at the 'word' tab, we can see that the precision and recall for category 1 (non-Frisian) were relatively low. There thus seems to be a bias in the classifier to predict category 0 too frequently. This bias could be the result of an unbalanced test set. This is unlikely as all sets are shuffled and then are divided in equal parts. It could happen that by chance more ambiguous messages happen to fall in the test set. It might also be a result of different forms of annotation; maybe the annotator for the Gronings set made less ambiguous choices for labels than the Frisian annotator did. As the expected 0.94 was at 'bag of words', it is likely that important words from the test set missed.

It would either way be a better idea to have multiple annotators per language in stead of one. Annotator bias could lead to reduced applicability of these classification models to external data. Another point of improvement lies in the Twitter query script that was run weekly. In this script, geolocation in the form of a circle in the provinces of Friesland and Groningen was specified. The choice for just one circle came from the fact that in this state, the script already used enough computational power. This way, Frisian and Gronings received a limited representation on twitter, using only one circle for each. Using more computer power to use multiple circles would have been a better choice, since this would vastly improve the full representation of Frisian and Gronings.

6 CONCLUSION

This research attempted to investigate the possibility of creating a high quality corpus of a dialect and minority language on Twitter and to use this corpus to identify Tweets in Gronings and Frisian from Dutch Tweets with high accuracy. In order to achieve this, the research question: 'is it possible to create a high-quality corpus of a dialect and minority language on Twitter and to then use this corpus to identify Tweets with high accuracy?' was formed. To answer this question, a set of sub-questions were formulated. The first sub-question was formulated as: 'To which extend is it possible to create a high-quality corpus from Tweets that contain the Gronings dialect, using seeding terms and geolocation'. After two months of querying a script, using geolocation and a growing list of keywords, an annotated corpus of 1000 unique messages in Gronings was developed. Every message in this corpus contained at least one word that was perceived to be Gronings by a native speaker. In addition to this, a accompanying set of 1500 messages that were identified as non-Gronings was identified. To answer the first sub-question, it is possible to create a high quality corpus using seeding terms and geolocation in the sense that a 1000 Tweets can be properly identified in a time frame of 2 months.

The second sub-question that was formulated was: 'To which extend is it possible to create a high-quality corpus from Tweets that contain the Frisian language, using seeding terms and geolocation'. This question can be answered in the same way as the previous sub-question. 1000 Tweets were equally identified by a native speaker in the span of two months. For Frisian, only 500 accompanying, non-Frisian Tweets were identified before the number of 1000 Frisian Tweets was achieved . We can thus conclude that identifying messages on Twitter that contain the Frisian language is easier than doing so for Gronings. It is possible to create a high-quality corpus from Tweets that contain the Frisian language, using seeding terms and geolocation, in the sense that a 1000 messages that contain the Frisian language were identified in the span of two months.

The final sub-question to answer the research question was formulated as: 'To which extent is it possible, with the created corpora, to correctly identify Tweets in the Gronings dialect and the Frisian language'. To answer this question, first two vectorization methods, count and TF-IDF, were compared on all classification models and tokenization types. Using the best performing results, classification models were compared with different tokenization methods were compared with each other. The result of this extensive comparison is that on the Gronings set, a SVM classifying data that was tokenized on a word level with TF-IDF performed the best. On this Frisian set, a naive Bayes model that classified data that was tokenized as words, without TF-IDF performed best. To answer the sub-question: a final test set comparison ruled that the Gronings corpus could be classified with a weighted f-measure of 0.93 and the Frisian corpus could be classified with a weighted f-measure of 0.83.

To answer the research question: 'Is it possible to create a high-quality corpus of a dialect and minority language on Twitter and to then use this corpus to identify Tweets with high accuracy?', the answer would be yes. It is possible to create a high quality corpus of 1000 annotated messages that contain (regional) language from both Gronings and Frisian, and with this corpus it is possible to identify Tweets with a relatively high accuracy of weighted f-measures of 0.93 for Gronings and 0.83 for Frisian.

BIBLIOGRAPHY

Johannes Bjerva Malvina Nissim Gertjan van Noord Barbara Plan Artur Kulmizev, Bo Blankers and Martijn Wieling. The power of character n-grams in native language identification.

Shane Bergsma, Paul Mcnamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific twitter collections. pages 65–74.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. *Proceedings of EACL 2009 Workshop on Computational Approaches to Semitic Languages*.

Jan Blommaert. 2012. The sociolinguistics of globalization. *Language Problems and Language Planning*, 36:114.

Alina Maria Ciobanu, Shervin Malmasi, and Liviu P. Dinu. 2018. German dialect identification using classifier ensembles. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 288–294, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

H. A. Elgabou and D. Kazakov. 2017. Building dialectal arabic corpora. *The Proceedings of the First Workshop on Human-Informed Translation and Interpreting Technology (HiT-IT)*, 59:52 – 57.

C.A.J. Hoppenbrouwers and G.A.J. Hoppenbrouwers. 2001. *De indeling van de Nederlandse streektalen: dialecten van 156 steden en dorpen geklasseerd volgens de FFM*. Koninklijke Van Gorcum.

Thorsten Joachims. 1998. Text categorization with support vector machines. *Proc. European Conf. Machine Learning (ECML'98)*.

M. Paul Lewis. 2009. *Ethnologue: Languages of the World, 16th edition*. Dallas Texas SIL International.

Nikola Ljubesic, Nives Mikelic Preradovic, and Damir Boras. 2007. Language indentification: How to distinguish similar languages? pages 541 – 546.

Bruno Martins and Mário Silva. 2005. Language identification in web pages. volume 1, page 764.

H. Mubarak and K. Darwish. 2014. Using twitter to collect a multi-dialectal corpus of arab. *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Langauge Processing (ANLP)*, pages 1 – 7.

Sam Scott. 2000. Feature engineering for text classification. *Proceedings of ICML 99, 16th International Conference on Machine Learning*.

Stephen Tratz. 2014. Accurate arabic script language/dialect classification.

Omar Zaidan and Chris Callison-Burch. 2013. Arabic dialect identification. *Computational Linguistics*, XXX.