Classification of the Original Author-Party of a Political Document Using a Multinomial Naïve Bayes Classifier

Anton Mo Eriksson, *TDDE16, Text Mining, January 21, 2019, anter491@student.liu.se, Linköping University – Linköping – Sweden*

Abstract—This paper outlines the procedure to scrape web API:s, data cleaning, and general text mining pre-processing, to achieve a $\mathcal{N}$-gram model of the scraped texts. All this to be able to classify the original author-party of a political document using a Multinomial Naïve Bayes classifier. The classifier performed with an accuracy of 41.5%, reasons for that are found in the discussion.

1 Introduction

Modern software engineering has given humanity the ability to work with big data and massive computational power offered by parallel computing in the cloud [1]. The massive expansion of the computational power introduces new domains and opportunities start to show themselves, machine learning and text mining are old ideas, which in recent years has been made plausible by the newly acquired computational powers [2].

Moreover, how should we use these newly acquired computational resources? Some would argue investment banking or smart city planning [3] [4]; however, an unexplored area is the political domain which will be the focus of this project. Then one may ask, what can we benefit from applying these computational resources to the area of politics. The answer to that question is quite simple, to use the full power of text mining to uncover what politicians and political parties *actually* stand for. All this to improve the democracy we live and thrive in, to transform our democracy from *democracy-1.0* to *democracy-2.0*.

In this project, the goal was to determent which Swedish political party was the original authors of a certain document. The concept of which party authored the document can reveal, interesting information regarding the possible political allies. This might be confusing to the politically engaged, which would claim only two solid political-blocks exist. However, shown by the recent shake-up of the Swedish government, there is a need for new thinking and innovation in this area. Furthermore, it also presents the opportunity to tie certain politicians within a party to certain policies, which can be of great interest to both the public as well as the party.

Finally, the task to classify the original author-party of documents collected from the open API[1] to investigate the unions and coupling within the Swedish parliament, boils down in the research question:

- *Can predictions be made of the original political faction based on the motion text, with a Naïve Bayes classifier?*

1.1 Limitations

The project will only focus on the party level, thus not the individual politician perspective, which is too broad of a scope for this project. But encourage the reader to peruse that area of research.

1. https://data.riksdagen.se/

2 THEORY

This section aims to give the reader some theoretical background to the technologies and methodologies used.

2.1 $\mathcal{N}$–gram model

The $\mathcal{N}$–gram model represents the sequence of $\mathcal{N}$ continues words from a text or a speech. The model aims to capture the conditional probability given by the $\mathcal{N}-1$ previous words, thus making the meaning of verbs more important to the model [5]. The $\mathcal{N}$-gram model can be concertized with any number, where $\mathcal{N}=1$ is known as unigram or Bag of Words, followed by bigrams for $\mathcal{N}=2$, where the probability model is described as the following:

$$P(w_1,..,w_{n-1}) = \prod_{k}^{\mathcal{N}} P(w_k|w_1,..,w_{k-1}). \quad (1)$$

2.1.1 Bag-of-Words

A special case of the $\mathcal{N}$–gram model is the unigram also known by the term, Bag of Words. The idea behind Bags of Words can be derived from the name, a dictionary structure of the occurrences of works, where the key is the word in question, and the value is the frequency. This Technique is good at capturing a representation of a string input, where the sting could be a comprehensive text, even of the use of a Bag of Words model can result in high dimensionality [6].

The procedure for Bag of Words can be described in the following steps; convert all alphabetical characters to lowercase and remove none alphabetic characters. Tokenize the text and then iterate through the text with the tokens as key and increment the frequency value.

2.2 Term Frequency-Inverse Document Frequency

The term frequency-inverse document frequency or (TF−IDF) is one of the most popular types of model in the text mining field [7]. It models the importance of a word in a given document collation, the TF−IDF is defined by two sub-formulas, term frequency and inverse document frequency , both presented below:

$$tf(t,D) = \frac{1}{2} + \frac{f_{t,d}}{2 \cdot \max\{f_{t',d} : t' \in d\})} \quad (2)$$

where: $tf_{t,d}$: the term frequency,
$f_{t,d}$: the raw frequency,
t : the count of the word,
t' : the most frequent word,
d : the document.

Followed by the second one,

$$idf(t,D) = log(\frac{N}{1+|d \in D : t \in d|}) \quad (3)$$

where: idf : the inverse document frequency
t : the count of the word,
N : the amount of documents in D,
d : the document,
D : the documents.

Those two formulas together defines the function for TF−IDF presented below:

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \quad (4)$$

The use of TD−IDF is an attempt to make words occurring frequently in few documents, not be over represented by the model [7].

2.3 Naïve Bayes

One of the most classic and old families of algorithms in the text mining field, but despite its age, it still packs a punch [8]. Thus, comparing equally or better than more advanced machine learning algorithms (depending on the area) [6]. Naïve Bayes in a probability classifier, using the famous Bayes theorem, with the prerequisite of independence between the features in the data set. The term naive is derived from the assumption of the independence between the features.

All the algorithms in the Naïve Bayes family label the input feature vector according to a set

of pre-defined labels. The procedure will be illustrated in the formula below:

$$\overline{\mathcal{L}} = arg\max_{c\in\mathcal{C}} P(c)\cdot\prod_{i=1}^{n} P(x_i|c) \tag{5}$$

where: $\overline{\mathcal{L}}$: the predicted labels
$\mathcal{C}$: the set of labels,
c : the concrete label,
n : the size of feature-vector,
x_i : i:th component in the feature vector.

There are several different variations of the Naïve Bayes classification [6], and thus in this project, the following has been chosen.

2.3.1 Multinomial Naïve Bayes

A variation on the Naïve Bayes is the Multinomial Naïve Bayes classifier, which differs on the assumption that the probability distribution is a multinomial [6], defined by the formula below:

$$P(w_1,...,w_n|c) \sim \mathcal{M}(\theta_{si}, n_i). \tag{6}$$

Hence, assuming that the distribution of the probability is multinomial. The formula for the Multinomial Naïve Bayes classifier is given by,

$$P(w_1,...,w_n|c) = \prod_{i}^{n} P(w_i|c_i). \tag{7}$$

One of the reasons for using the Multinomial Naïve Bayes is due to research presented by Kibriys *et. al* which pointed out the great result from a Multinomial Naïve Bayes classification in the area of text classification [6].

2.4 Metrics

The following metric was used to evaluate and validate the output from the used classifier.

2.4.1 Accuracy–Score

The accuracy is the number of correct predictions made by the classifier divided by the total number of predictions [9], illustrated in the formula below:

$$\mathcal{A} = \frac{tp+tn}{tp+tn+fp+fn} \tag{8}$$

where: fn : false negative,
fp : false positive,
tn : true negative,
tp : true positive.

Despite seeming like a precise metric, accuracy can be misleading if it is the only metric used, due to if tn is large the result will be twisted.

2.4.2 Recall–Score

This metric describes the ability of the classification to label the correct with missing [9].

$$\mathcal{R} = \frac{tp}{tp+fn} \tag{9}$$

2.4.3 Precision–Score

This metric is an indication on how well the classification has performed regarding correct classification divided with the total classification of that class [9].

$$\mathcal{P} = \frac{tp}{tp+fp} \tag{10}$$

2.4.4 F1–Score

The F1 score is a combination of the prediction and recall in one of Pythagoras mean (harmonic mean) between the two [9].

$$\mathcal{F}_1 = \frac{2\cdot\mathcal{P}\mathcal{R}}{\mathcal{P}+\mathcal{R}} \tag{11}$$

2.4.5 Confusion Matrix

This is a metric design to give a good overview of the result, where high numbers along the diagonal represent desirable results. An example of a confusion matrix is illustrated in table 1.

TABLE 1
Example of confusion matrix.

		Expected	
		True	False
Predicted	True	tp	fp
	False	fp	tf

N

3 DATA SET

The data set in question is a scraping of the API[2] provided by the Swedish parliament. The API provides a number of different format for the API request, e.g. JSON, HTML with more. In this project HTML has been the favored data transfer format, due to the helpfull Python[3] library BeautifulSoup[4], which parses HTML.

In the files from the API in question, where political motions, from any party in the Swedish parliament, thus giving eight unique parties, which wear used as label corresponding to the respective document.

Furthermore, the pre-processing and data representation, some Python libraries have been used to develop the classification system, the list of the most important follows below:

- Pandas[5] – An open source data analysis library, which provides excellent data structures to use in text mining.
- scikit[6] – Open source data mining and analyzing library, where the project classifiers are imported from.
- SpaCy[7] – An natural language processing liberty.
- BeautifulSoup – Great library for HTML parsing.
- Pickle[8] – A liberty for serialization of data structures.

4 METHODOLOGY

The project can be divided into three distinct sub-sections *data collation*, *pre-processing* and *training and analyses*.

4.1 Data Collation

The data collation process began with the task of acquiring the necessary amount of political document to perform text mining. The area of *motions* was selected, this because motions should contain concrete politics reflecting the parties opinions. The scrapping for documents of the type motions in the search API[9]. All motions can be listed and from which the HTTP requests yields a HTML version of the original PDF or docx, which could be extracted from the API.

To avoid making the HTTP request on every execution the collected data was saved with the help of the Python library Pickle which makes it easy to save and load the Python data structure e.g. Dictionary.

4.2 Pre-processing

Pre-processing was one of the most difficult and import areas of the project, to clean up the inconsistently structured data to be able to get reasonable classification results. The data cleaning was done in different steps, first the extraction of the plain-text from the HTML version of the motions and the corresponding labels of the text, the author party. To extract the author party prove to be one of the troublesome thing in the project due to the inconsistency of the structures in the provided files, e.g. the party was mostly labeled as ”(V)” or ”(MP)” but in some cases ”(bada M)” and sometimes at multiple places, which made it tough to create robust and trustworth regular expression to extract those.

When the text and the party labels had been extracted, the task to remove numerical values, stop words and converting it to lowercase where conducted. In regards to stemming of the data, recent studies by Schofield *et. al* have indicated that stemming can produce a worse result compared to none-stemming, so it was left out [10].

2. http://data.riksdagen.se/
3. https://www.python.org/
4. https://www.crummy.com/software/BeautifulSoup/doc/
5. https://pandas.pydata.org/
6. https://scikit-learn.org
7. https://spacy.io/
8. https://docs.python.org/3/library/pickle.html
9. http://data.riksdagen.se/dokumentlista/

4.3 Training & Analysis

Lastly, in the training and analysis the 1700 documents collected in the scrapping where divided in to test and training data (20% to 80%). Then from the training data a Multinomial Naïve Bayes classifier where trained with the unigram Bag of Words data structure as input. The prediction from the classifier then where displayed with 5 different metrics; accuracy, recall, precision, F1, and confusion matrix. All this to give as much insight about the classifier as possible.

5 Result

The data collation processes yielded 1700 number of political motions, authored by one of the eight Swedish parliament parties the distribution among them will be presented in the table below:

TABLE 2
Distribution of the collected motion among the parties.

Party	V	S	MP	SD	M	L	C	KD
Amount	53	334	57	249	567	165	209	66

Moreover, the res

5.1 Metrics

5.1.1 Confusion Matrix

The generated confusion matrix by the classifier will be illustrated in table 4.

6 Discussion

With the result in mind, we can see that a Multinomial Naïve Bayes classifier performed rather poorly in regards to the five applied metrics. One possible error source for the classification is the problematic task to acquire enough data to satisfy the general case fitting. The problem with acquiring enough data has its roots in the poorly structured `API` of the Swedish parliament, where the lack of consistency between the documents puts a pin in the wheel of effective document scrapping. However, that comes as no surprise, because there is nothing but the Swedish principle of open government that forces the availability of the documents.

TABLE 3
The resulting values of the applied metrics to the classifier.

Metric	**Score in %**
$\mathcal{A}$	41.5
$\mathcal{R}$	41.5
$\mathcal{P}$	57.2
$\mathcal{F}_1$	36.6

TABLE 4
Resulting confusion matrix.

42	60	0	71	0	0	2	0
0	**32**	0	14	0	0	3	0
1	62	**36**	37	0	0	0	0
4	101	0	**353**	0	0	0	0
0	22	0	21	**0**	0	0	0
7	124	0	114	0	**17**	2	0
0	83	0	86	0	0	**24**	0
0	24	0	13	0	0	0	**5**

Continuing to the research question for the project, which has shown poor results the classification of the original party, identification of possible reasons for that lies in the close coupling within the Swedish parliament which makes it even hard for humans to derive the original author only by text. As shown by the confusion matrix for the classifier. So, to try to answer the research question, it would be hard to argue for the classification process at this time with the poor results in mind; still the classification is correct roughly every other time. Thus, it would indicate a future possibility to use a Multinomial Naïve Bayes classifier to predict the author of a document, with higher scores on the measured metrices.

Furthermore, possible improvements areas have been identified as, more training and testing data, at this point the document count is 1700 which is roughly half of the motions form period 2017-2018. Moreover, a different variation of $\mathcal{N}$-gram should be tested and evaluated to use the $\mathcal{N}$ with the best performance. Lastly, a trail with a different

classification model would bring new insight to the project.

7 Conclusion

The conclusions that can be derived from this project is the importance of coherence in the data, which will be used in the project. In this case, the data mined from the Swedish parliament `API` contained inconsistency, which required much time and effort to clean up.

Lastly, the use of classification in the area of politic, more special predicting the author of a document could be a great tool for the future if this approach where developed beyond the scope of this course. Finalizing with the reflection that with more time and effort the classification of document author is a possibility.

References

[1] T. Lin and S. Wang, "Cloudlet-screen computing: a multi-core-based, cloud-computing-oriented, traditional-computing-compatible parallel computing paradigm for the masses," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pp. 1805–1808, IEEE, 2009.

[2] T. G. Dietterich, P. Domingos, L. Getoor, S. Muggleton, and P. Tadepalli, "Structured machine learning: the next ten years," *Machine Learning*, vol. 73, no. 1, p. 3, 2008.

[3] R. Choudhry and K. Garg, "A hybrid machine learning system for stock market forecasting," *World Academy of Science, Engineering and Technology*, vol. 39, no. 3, pp. 315–318, 2008.

[4] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proceedings of the ASE BigData & SocialInformatics 2015*, p. 28, ACM, 2015.

[5] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[6] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial naive bayes for text categorization revisited," in *Australasian Joint Conference on Artificial Intelligence*, pp. 488–499, Springer, 2004.

[7] S. Vijayarani, M. J. Ilamathi, and M. Nithya, "Preprocessing techniques for text mining-an overview," *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2015.

[8] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[9] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[10] A. Schofield, M. Magnusson, and D. Mimno, "Understanding text pre-processing for latent dirichlet allocation," in *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics*, vol. 2, pp. 432–436, 2017.