NLP approach to Annual Reports Analysis

Quintana Pelayo, Guillermo
AI Master Student
Data Science and Knowledge Engineering
Maastricht University
g.quintanapelayo@student.maastrichtuniversity.nl

27 May 2020

Abstract

The internet is full with financial information about companies available to investors. There has been several attempts to make predictions over stock prices and extract relevant information from news articles, however, it has been difficult to extract meaning from the textual parts of annual reports automatically. These texts contain valuable information in a more complex structure and vocabulary than a regular article. The aim of this paper is to make use of text mining methods in order to extract relevant information about changes in the company, such as people, new products or stores, from its annual reports as well as extracting the general sentiment of these texts. All these methods were tested with Tesla annual reports, and they proved to show valuable and interesting information not visible at plain sight. The system presented here could be a good start towards a more complete and powerful software for investors.

This project is part of the Information Retrieval and Text Mining course from the Faculty of Data Science and Knowledge Engineering Master program of the Maastricht University.

Keywords: *NLP, text mining, information retrieval, annual reports*

1 Introduction

There has always been an interest in automatic company analysis based on natural language processing, this interest has increased over the past years particularly with Quantitative value investing [1], which seeks the Holy Grail formula or system capable of analyzing fundamental data. One important reason for this interest is to provide investors another source of market intelligence in order to make the right decisions. Thus, many attempts have already been done in this area with the promise of better and fast market insights without the need of well trained analysts going though all the files published by a company. Even though the final purpose of a system like this would be very bounded to stock market prices, this paper moves away from that and focuses only on the data structure and analysis, meaning future financial performance predictions are out the scope of the present project.

This paper focuses on three main objectives, firstly, **extract valuable information** about relevant people (such as employees or a CEO change) and products. Secondly, detect the **general sentiment** of the text: the company may be explaining stuff more cautious due to possible setbacks, or being optimistic about the future. And lastly, obtain an auto generated **summary** with the most relevant topics for each document.

The paper is structured as follows. Section 2

introduces the previous work that has been done in this area. Section 3 describes the methodology investigated. Experimental results are reported in section 4, and conclusions and future work are stated in section 5 and 6, respectively.

2 Previous work

The idea presented here isn't new, actually it has been studied for more than five decades but with different points of view. Following a chronological timeline, first studies that stated showing interest in the analysis of annual reports or other financial publications focused in detecting the honesty of these companies that may be trying to cover up their records with obscure language.

Several studies have researched the clarity and honesty of the reports and, in particular L.D. Parker among others, found that these texts showed a high reading level similar to the more technically written financial statements [2] and that the readability of the reports did not appear to improve noticeably over time [3].

Moving onto a closer approach, Mittermayer [4] presented *NewsCATS* in an attempt to automate the trading decisions based on news articles immediately after they are released. Afterwards, categorize news articles using a standard three tags convention (good, neutral and bad) in classification problems and then relate each article to stock index variations.

As it is logical, Machine Learning and Natural Language Processing techniques soon reached financial predictions. Butler [5] combined Readability functions and Word N-grams with a bag-of-words approach to obtain an average accuracy of 69%. Falinouss [6] focused on intra-day stock prices predictions using vector space modeling, tfidf term weighting scheme and Support Vector Machines which obtained a notable accuracy: 83%.

Probably the most investigated method in this field is sentiment analysis in financial news [7, 8, 9, 10]. It aims to understand how this news influence the investors decisions; unfortunately, there is no way to exactly know how that traders react to information released in the newspapers columns, "When you see it on the Wall Street Journal, it's already too late" - The Wolf of Wall Street. Goryachev et al. [11] related to negation detection in sentiment analysis proved that "NegExpander" resulted in the best Kappa values for all the experiments they conducted and they conclude saying "regular expression and syntactic processing based algorithms have better agreements with human reviewers than the classification-based algorithms".

Financial reports have also been clustered for quantitative analysis, like Wang [12] or Kloptchenko [13], and leaving apart the stock prices prediction as the present project. Here, the authors used prototype-matching text clustering and collocational networks to visualize the reports. Improved later in [14] with self-organizing maps. Some indication about the financial performance of the company can be gained from the textual component of the reports. IN both cases the clusters from quantitative (past performance) and qualitative analysis did not coincide.

Finally I'd like to acknowledge the work from Turegun [15] that provided me a good ground based introduction to financial text mining as well as a long list of previous work done in this field.

3 Methodology

This section describes the techniques and steps followed. Everything has been coded using Python3 and also with the help of the Linux bash for data and files handling.

All the data is publicly available in PDF format, which means that the first step required is to transform everything into plain text. This transformation can lead to inconsistencies in the text and, in many cases, information loss such as figures, tables or any other graphic representation. Thankfully an annual report needs to follow some aesthetic guidelines to avoid flamboyant and unnecessary representations. Thus, we need to make use of a graph matching technique to wrap the text such as python's library *tika* (following the work of Hassan et al.[16] and Oro et al. [17]) and the builtin *codecs*, the PDFs were transformed into plain text files with no formatting. It's important to notice that this is the only step where the data files were treated as binaries, afterwards, everything will be written and read directly in text format using *UTF-8* encoding.

3.1 Preprocessing

Every NLP application requires this preprocessing step to prepare the data, text in this case, that is going to be used later on. This section explains the steps necessary to perform this preparation. The operations are scheduled in a common pipeline for this kind of projects, which is: normalization → tokenization → sentence duplication → POS tagging.

Normalization

First of all, the text is *normalized* by erasing any special character: ©, ·, $, €, etc. Then, all parenthesis and its content as well as any URL or link inside the document. The final sentence dots are substituted by the keyword "END_OF_LINE" to facilitate future sentence splitting. We have to keep in mind that this normalization creates a loss in structure and semantic meaning, thus, the output sentences in the postprocessing will have no structure at all and will be hard to read. Nevertheless, this loss is necessary for the next steps and in order to obtain the best results possible.

Only the text summarization requires to go back to this normalization and perform two additional changes: lowercase all the text and remove all digits. This is not the case for all the postprocessing pahses since the capitalization of the very first letter of some words helps to identify certain entities such as names and the numbers are not a problem to worry for now but could be useful in the future.

Tokenization

For the tokenization process I dedided to stick with M. Hassler and G. Fliedl work [18], which is a very robust technique shown here in algorithm 1. I wont go into details since its paper well explains the process to follow but I'd like to mention that I modified the such named "end markers" that are now precisely blanks, tabs, new lines and line feeds specified with their UTF-8 identifiers.

Sentence Duplication

The goal of this step is to detect similar sentences inside a document in order to erase unwanted legal text that every company needs to include in their annual reports.

To achieve this, Efstathiades et al. [19] uses

Algorithm 1: Tokenization and typing of tokens from M. Hassler and G. Fliedl 2006

Data: Plain text
Result: Tokenized text set

```
begin
    identify single-tokens
    type single-tokens
    split sentence end markers
    reinterpret single-token types
    merge and split tokens recursively
    reinterpret all token types
end
```

"Link of Interest" (LOI) and "K Relevant Nearest Neighbor"(K-RNN) to obtain the relevant location point of interest in a query or in this case a sentence. Using this way of relevance scoring and then tuning a "sensitivity" parameter was, at first sight, a good solution. At the end I couldn't get to work the LOI and K-RNN to compute the similarity between sentences but I realised just using Levenshtein Distance turned out to be a quick and reliable solution and the number of sentences in the annual report text files suffered a reduction of 27% on average. Reduction that is indeed very helpful when it comes to reducing the total execution time of the postprocessing algorithms and classifiers.

Part Of Speech (POS) Tagger

The last aspect to cover in this pipeline is the POS tagger that will be then used for the first attempt in Named Entity Recognition. The tagger used was originally written by Kristina Toutanova, an implementation of the log-linear part-of-speech taggers described in this paper [20], that achieved an impressive 97.16% accuracy which ensures that this tagger isn't a bottleneck of wrong tags assigned that could lead to a worst performance later on.

3.2 Postprocessing

Once the data is ready to perform operations and analysis with it, we now move to the postprocessing part of the present project; which relies on three different applications independent from each other: Name Entity Recognition (NER), Auto summarization and Sentiment Analysis.

Named Entity Recognition (NER)

Since we already have a tagged corpus for each document, the last step required before doing NER is to make use of a dependency parser to assign dependency labels to each word in every sentence and then use any classification method to train and predict entities on our text. The first thought was to implement a deterministic dependency parser based on the work by Nivre and Scholz [21], which parses English text in linear time and showed a good overall accuracy of 86% when restricted to gramatical role labels. This approach uses Memory-Based Learning which reuses solutions from previously solved problems, however, when trying to recreate it using data from the Penn Treebank and IB1 algorithm [22], the resulting accuracy was always between 53% and 58%. I considered this isn't good enough so I moved onto a pre-trained model that can both perform the dependency parsing and the NER. I can then apply this model to the present problem, in particular using a convolutional neural network (CNN).

The selected multi-task pre-trained CNN to perform the NER classification was

"en_core_web_sm" in his latest version 2.2.5, which detects and classifies 18 different tags: CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART. This CNN is very useful for several main objectives (detect relevant people, places and product announcements), it has been trained with more than 650k English news, meaning it is a general classifier instead of a specific economy related dataset. Even though it classifies that variety of entities, the most important ones that were used here are this three:

- *PERSON*: People, including fictional.
- *PRODUCT*: Objects, vehicles, foods, etc. (Not services.)
- *ORG*: Companies, agencies, institutions, etc.

Because this model is statistical and strongly depend on the examples it was trained on, this doesn't always work perfectly and might need some tuning, which it's discussed in section 4.

Auto summarization

The auto summarization wasn't part of the initial goals of this project but it turned out to be quite important given the length of annual reports. These summaries must keep the overall idea of the text but it is forgivable if it misses a few important products or events since they will be extracted with the above mentioned entity recognition.

In order to obtain a summary for the annual reports we'll make use of TextRank [23], a variation from the popular PageRank.

This basic but useful automatic text summarization has this workflow: remove stop words → build a similarity matrix using cosine similarity→ generate rank based on matrix → pick top N sentences for summary.

Sentiment analysis

As stated in section 2, many attempts have been done regarding sentiment analysis in financial texts. I opted to try an approach from the bottom and classify sentences into three categories: positive, neutral and negative. The first step was to collect already labeled text with these categories, here I made use of a completely off-topic dataset but very large consisting on 1.6 million tweets and also another smaller one (4800 entities) but constructed using financial articles headlines. Then I trained these datasets separately with different popular classifiers (logistic regression, naive bayes, support vector machines, K nearest neighbours, stochastic gradient descent and neural networks) using 10-fold cross validation. Table 1 shows the training area under curve (AUC) results measured and the different parameters used depending on the algorithm.

Section 4 will reveal that these datasets didn't perfom very well predicting over the annual reports. That prediction validation required manually labeling sentences from annual reports so I manually labeled 1100 sentences to positive, neutral and negative extracted from Tesla 2010 report according to my own subjective thoughts (not very scientific but good enough for testing purposes). After this attempt I combined the financial articles dataset with the one that I created myself with these sentences and it had a training AUC of 98% using SGD, even more than the previous best.

Classifier	Parameter	FN	Tw
Log Reg	C=1	0.92	0.89
Log Reg	C=0.8	0.9	0.86
Log Reg	C=0.5	0.86	0.72
Log Reg	C=0.2	0.76	0.68
Naive Bayes	-	0.92	**0.91**
SVM	k=linear	0.96	-
KNN	-	0.89	-
SGD	-	**0.97**	0.82
NN	epoch=5	0.94	0.88

Table 1: AUC results for every classifier tested. FN = financial articles dataset, Tw = tweets dataset. Highlighted values are best for that particular dataset. Missing results '-' were unable to obtain due to the size of the dataset.s

4 Experiments and Results

The experiments were conducted in a company from the SP500 stock market, Tesla. This company was selected due to personal interest and because it has made recent progress in the area of artificial intelligence and machine learning; so it fits perfectly with the scope of this work.

4.1 NER

The entity recognition reveals very interesting insights over Tesla as can be seen in figure 1, specifically it's quite relevant the importance *Elon Musk* is given by its company (bottom graph). He maintains the biggest mentions inside these annual reports apart from the one from 2015, surpassed by *Jason Wheeler* Chief Financial Officer (CFO) which left Tesla that same year to pursue opportunities in public policy. Previously this position at Tesla was held by *Deepas Ahuja*, whose importance is also clearly visible until 2015. In terms of products (top graph) it can be appreciated a reduction over the focus in the *Tesla Roadster* and the *Model S* towards the *Model 3* and the starting rise of the *Model Y* and the *Cybertruck* only mentioned on the last report. It also reveals the appearance of the *Solar Roof* as a product after the acquisition of Solar City.

I found specifically significant that the NER tagged the *Supercharger* and the *Autopilot* as products, I wouldn't think about them as products but it makes sense that the company talks about them as if they were since they are services offered by the company.

In terms of the system measures table 2 shows the obtained results for the best and worst classified Tesla's annual reports, overall the precision stays over 0.60 apart from year 2013, a possible explanation is that this document turns out to be the longest and also the classification errors here occurred mostly over the products tags (example included in appendix figure 4).

Year	Prec	Recall	f-score	Kappa
2010	0.80	0.99	0.89	0.79
2013	0.53	0.98	0.69	0.50

Table 2: Precision, recall, f-score, and kappa values (compared to random) from best (2010) and worst (2013) classified documents by the described NER.

4.2 Summary

The execution time is significantly high (2h) since the similarity matrix dimension is above 2000x2000 instances in most of the cases. Furthermore, the length of every sentence is also quite large, reaching in some cases 120 words per sentence. Nevertheless the summaries gen-

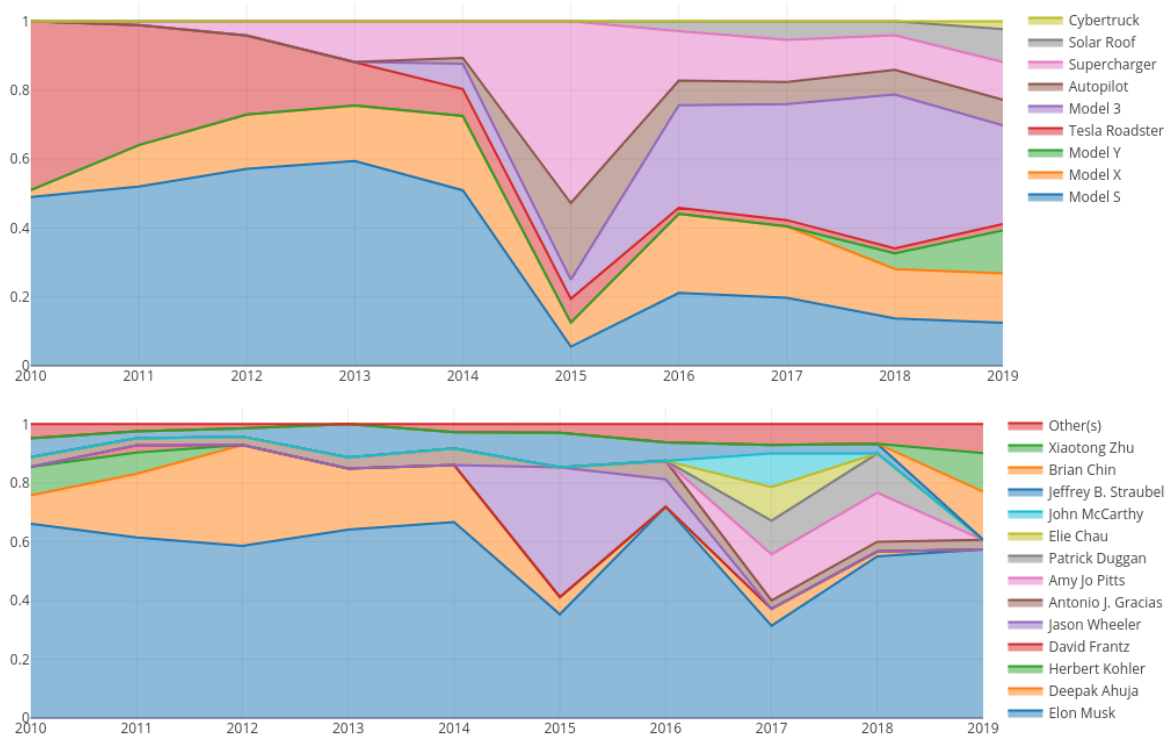

Figure 1: Area representation of the discovered entities by their percentage of appearance in the text (y axis) and the year of the document (x axis). The entities are divided into products (top) and people (bottom) from Tesla annual reports, each color represents a different entity, the wider the area is, the more it appears on the document.

erated by the system actually contain key interesting aspects, figure 2 has the first 3 sentences returned by the algorithm sorted by score using TextRank.

4.3 Sentiment analysis

Table 3 contains the results classifying sentences for the 2011 Tesla annual report, the results aren't excellent but the increased financial news dataset with the manually labeled sentences shows potential to be improved in the future adding more instances since now only consists in almost 6000.

And the final sentiment classification obtained can be seen in figure 3, the only remarkable change that can be spotted is a more prudent explanation since 2016 proved by the augment of neutral sentences over positive/negative. Also since its the company itself who is writing the text, it's no surprise the huge amount of positive occurrences; which leads to the conclusion that a light increase in negative clauses should be taken much more into account than large variations in the positives.

* in addition until june before licensing intellectual property generated outside the scope of any strategic cooperation area to a daimler competitor we would first have to offer dnac the right to license the intellectual property on a non exclusive royalty bearing basis or on an exclusive basis in the automotive field and if dnac requests the latter we must negotiate such a license in good faith

* we may not be able to potential delays in launching the model s if we lose daimler s automotive support and are unable to find an alternative in a timely manner

* potential loss of access to various parts that we are incorporating into our model s design and potential loss of business and adverse publicity to our brand image if there are defects or other problems discovered with our electric powertrain components that daimler has incorporated into their vehicles

Figure 2: First 3 sentences selected from Tesla 2010.

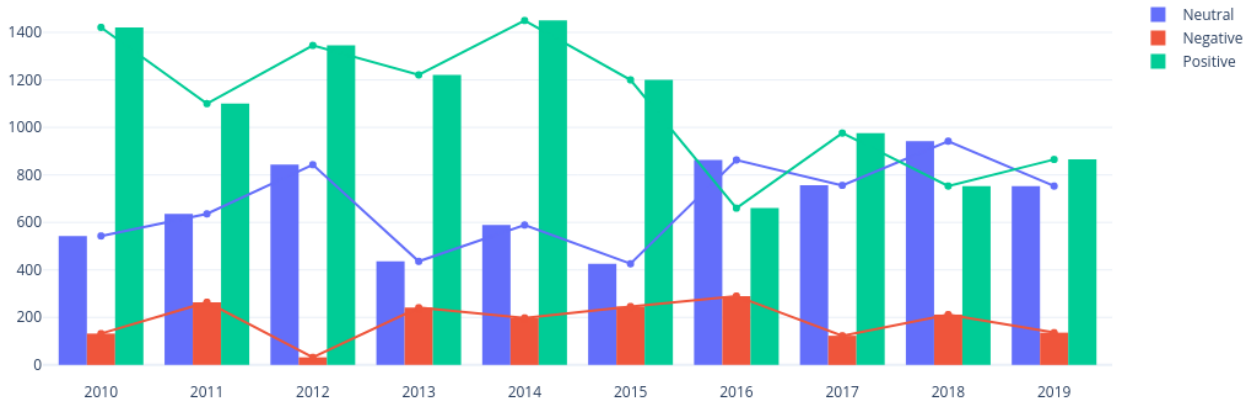

Figure 3: Area representation of the discovered entities by their percentage of appearance in the text (y axis) and the year of the document (x axis). The entities are divided into products (top) and people (bottom) from Tesla annual reports, each color represents a different entity, the wider the area is, the more it appears on the document.

5 Conclusion

This work has explored the application of natural language processing techniques to annual reports from Tesla. Given the complexity of working with unlabeled data and technical language, the results obtained are fairly respectable. And although the sentiment analysis accuracy falls short of the best available classifiers, the named entity recognition and the auto summarization appears to be promising. We also have seen that

Classifier	DS	Prec	Rec	f-score
Log Reg	Tw	0.53	0.54	0.52
Naive Bayes	Tw	0.58	0.63	0.56
Log Reg	FN	0.40	0.42	0.37
SGD	FN	0.42	0.51	0.40
SGD	**FN+**	0.69	0.78	0.68

Table 3: Precision, recall and f-score from the best classifiers for each dataset (plus Logistic Regression for comparison). FN+ = Financial articles dataset plus manual tagged sentences from Tesla 2010.

relevant information has been revealed when it comes to the people and products mentioned in the documents accomplishing the initial main goals.

6 Future work

This project had many personal interest and it has been very enlightening. I would like to continue expanding and completing this project into a more complex analysis and exploring more the possibilities of such system. In particular, this work still has a lot of room for improvement mainly in the sentiment analysis, were several sentiments could be used for classification instead of the three tags used and negation handling like the one used in [24]. Also, it could combine more documents related to a company in order to have both the opinion that the company wants to give to the customers and investors as well as the public opinion of newspapers and journalists. An individual person analysis could be added such that we can also obtain automatically additional information about the people mentioned in this documents as well as the study over more entity tags. I'd like to use this work as a base for my final thesis of the master.

References

[1] Wesley R Gray and Jack R Vogel. *Quantitative Momentum: A Practitioner's Guide to Building a Momentum-Based Stock Selection System*. John Wiley & Sons, 2016.

[2] NR Lewis, LD Parker, GD Pound, and P Sutcliffe. Accounting report readability: The use of readability techniques. *Accounting and Business Research*, 16(63):199–213, 1986.

[3] LD Parker. Corporate annual reporting: a mass communication perspective. *Accounting and Business Research*, 12(48):279–286, 1982.

[4] M-A Mittermayer. Forecasting intraday stock price trends with text mining techniques. In *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, pages 10–pp. IEEE, 2004.

[5] Matthew Butler and Vlado Kešelj. Financial forecasting using character n-gram analysis and readability scores of annual reports. In *Canadian Conference on Artificial Intelligence*, pages 39–51. Springer, 2009.

[6] Pegah Falinouss. Stock trend prediction using news articles: a text mining approach, 2007.

[7] Paul C Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3):1139–1168, 2007.

[8] Elaine Henry. Are investors influenced by how earnings press releases are written? *The Journal of Business Communication (1973)*, 45(4):363–407, 2008.

[9] Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467, 2008.

[10] Elaine Henry and Andrew J Leone. Measuring qualitative information in capital markets research: Comparison of alternative methodologies to measure disclosure tone. *The Accounting Review*, 91(1):153–178, 2016.

[11] Sergey Goryachev, Margarita Sordo, Qing T Zeng, and Long Ngo. Implementation and evaluation of four different methods of negation detection. Technical report, Technical report, DSG, 2006.

[12] Baohua Wang, Hejiang Huang, Xiaolong Wang, and Wensheng Chen. An ontology-based nlp approach to semantic annotation of annual report. In *2009 International Conference on Computational Intelligence and Security*, volume 1, pages 180–183. IEEE, 2009.

[13] Anotonina Kloptchenko, Camilla Magnusson, Barbro Back, H Vanharanta, and A Visa. Mining textual contents of quarterly reports. *Turku Center for Computer Science Technical Reports*, 2002.

[14] Antonina Kloptchenko, Tomas Eklund, Jonas Karlsson, Barbro Back, Hannu Vanharanta, and Ari Visa. Combining data and text mining techniques for analysing financial reports. *Intelligent Systems in Accounting, Finance & Management: International Journal*, 12(1):29–41, 2004.

[15] Nida Türegün. Text mining in financial information. *Curr. Anal. Econ. Finance*, 1:18–26, 2019.

[16] Tamir Hassan and Robert Baumgartner. Using graph matching techniques to wrap data from pdf documents. In *Proceedings of the 15th international conference on World Wide Web*, pages 901–902, 2006.

[17] Ermelinda Oro and Massimo Ruffolo. Xonto: An ontology-based system for semantic information extraction from pdf documents. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 118–125. IEEE, 2008.

[18] Marcus Hassler and Günther Fliedl. Text preparation through extended tokenization. *WIT Transactions on Information and Communication Technologies*, 37, 2006.

[19] Christodoulos Efstathiades, Alexandros Efentakis, and Dieter Pfoser. Efficient processing of relevant nearest-neighbor queries. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 2(3):1–28, 2016.

[20] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language*

technology-volume 1, pages 173–180. Association for Computational Linguistics, 2003.

[21] Joakim Nivre and Mario Scholz. Deterministic dependency parsing of english text. In *Proceedings of the 20th international conference on Computational Linguistics*, page 64. Association for Computational Linguistics, 2004.

[22] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[23] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[24] Isaac G Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 51–59. Association for Computational Linguistics, 2010.

Appendix

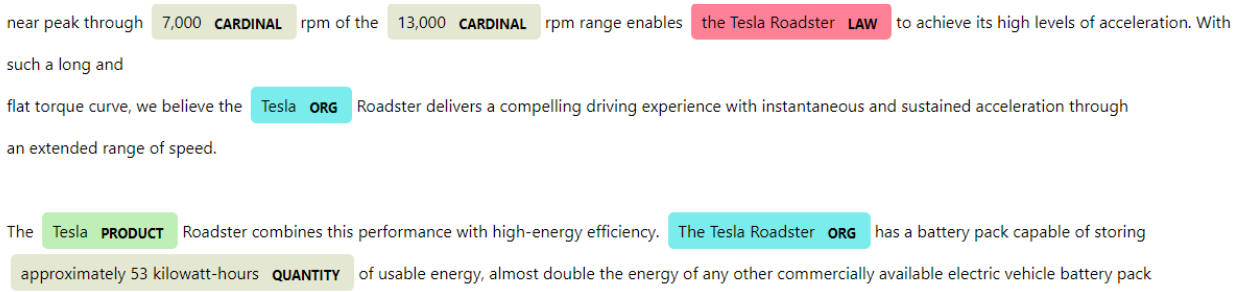

Figure 4: "Tesla Roadster" is classified here in 4 different ways.