Business Intelligence Capstone
Deep News Tagging with Dun & Bradstreet

Yuhan Yin, Jason Petri, Christian Lee, Shan Ali, Nai Wang

May 15, 2021

1 Executive Overview

Discovering value from unstructured data sources is challenging, but the difficulty is subsiding with strides in technological advancement in the natural language processing field. Under the supervision of associates at Dun and Bradstreet, we were tasked with uncovering value from business-related news. Our analyses involved creating classifiers to determine whether news articles are business or non-business in nature. From these determinations, further classifiers would be applied to business-related news. We built a multiclass classifier to determine, from a list of relevant subjects, what type of business news was described in the articles. One subject in major subject in question was merger and acquisition related news. Once classified as merger and acquisition related, we employed methods to extract who was the buyer, seller, the target, and the transaction price. These models would be used as a pipeline to generate insight for mentioned companies from raw, unstructured data.

2 Articulation of Business Outcomes

2.1 Business Context

This capstone project was sanctioned by a data, analytics, and insight company named Dun & Bradstreet. This company delivers data of many types to its stakeholders, including information about public and private companies. Tracking thousands of private and public companies is very time consuming, if completed manually, but can become a very valuable product to offer. This project addresses these inefficiencies in data collection. We attempt to automate the process to go from unstructured public announcements of various business news to structured, company specific information.

2.2 Business Questions

The motivation behind this project is to determine if advanced predictive modeling could be applied to unstructured text data to alleviate human components of data collection processes. Specifically, this task poses the following questions:

1. Can a model determine the difference between a news article that contains either business related news or non-business related news?
2. Of business classified news, can a model determine if the article pertains to specific subjects relevant to business?
3. Of subject classified news, can a model identify facts articulated throughout the article?
4. Can companies mentioned in articles be identified and assigned to an identification number for future pairing of data?

These questions are highly relevant to one another. These questions are chronologically ordered in terms of how succsessive predictive models would be applied in order to extract value. This value will come in the form of applying insights gained from the unstructured data source to a larger, structured database of company information.

2.3 Business Outcomes

The client, or the associates at Dun and Bradstreet governing this project, can expect a two part classification pipeline. This pipeline will take text from any news article and determine if it is business-related news or non-business news. If the article is related to business, our model will then determine if the article pertains to one of the following subject categories:
Funding; Earnings; Tradeshow; Real estate; Personnel; Awards; Contracts; Corporate expansion; Acquisitions, mergers, and takeovers; Partnerships; Dividends; Stock-offering; New products; Investment opinions; Surveys, polls, and research; or Bankruptcy.

From these classified articles, we can extract pertinent information related to mentioned companies and the article subject. For example, for the acquisitions, mergers, and takeovers article type, we can use models to answer questions like:

- What company was the buyer?
- What company was the seller?
- What company was the target?
- How much was the target acquired for?

With a completed pipeline and answered questions, this information would then be applied to Dun and Bradstreet's internal databases to offer alongisde their existing services. Assessing the numeric value is difficult without internal information. Therefore, one could reason the following two areas of value generation: optimizing existing processes or offering a new product entirely.

Internally, these models could feasibly replace many human elements for data collection. These models can classify news very fast, compared to an hour of labor. We estimated that classifying an article as business or non-business would take a human roughly four seconds. This equates to 900 news articles per hour. All 900 of these articles could be classifed by our models in roughly thirty seconds with high degrees of accuracy. Clearly, a human element could be removed thus promoting cost savings.

Additioanlly, the data generated from selling company-related data of this sort can be considered difficult to obtain. Therefore, this presents an business opportunity for Dun and Bradstreet to offer to new or existing clients. The exact value of this data is subjective and depends on the client's applications, but similar services can be priced in the thousands of dollars.

3 Commentary on Data Sources

The analysis for this project was conducted on two separate datasets. Each of the datasets pertained to a specific classification problem. The first dataset was for our binary classifier that we tasked with determining business related news from non-business related news. The second dataset was for our multiclass classifier that was tasked with determing what the subject of the article was from a predetermined list of categories.

3.1 Binary Dataset

This dataset contains 6,731 news articles. For each news article, it is labeled as relating to the financial markets, general business, technology, politics, entertainment, sports, or the world. For the purposes of this classifier, we are only concerned with articles that are labeled as dealing with the financial markets, business, and technology. These articles are labeled "business" and otherwise "non-business." This results in 968 business news articles and 5,764 non-business articles.

All news in this dataset contain the class label, the text of the article's headline, and the text of the article's body. In Figure 2, you can observe the distribution of classes for this classification problem.

With these two classes, it is important to maintain balanced classes while training any models. Any imbalance could have an adverse impact on observed results that skew the success or failure of a model's performance. All further data engineering will be addressed in the respective model explanations later in this report.

3.2 Multiclass Dataset

This dataset contains 17,470 news articles. For each news article, it is labeled as one of the following classes: Funding; Earnings; Tradeshow; Real estate; Personnel; Awards; Contracts; Corporate expansion; Acquisitions, mergers, and takeovers; Partnerships; Dividends; Stock-offering; New products; Investment opinions; Surveys, polls, and research; Bankruptcy; or Obituaries. For our purposes, the obituaries class has been removed. The removal of obituaries leaves 17,433 news articles. All news in this dataset contain the class label, the text of the article's headline, and the text of the article's body.

Figure 3 shows the balance of the data across classes. The balance of classes is important while conducting classification problems. Balancing of data should be attempted prior to training any models. All further data engineering will be addressed in the respective model explanations later in this report.

4 Analyses

Our analysis is compartmentalized in three major chronological steps:

1. Binary classifier—business or non-business news
2. Multiclass classifier—classify news category
3. Extract insight from classified news

Each step of analysis calls for different methods and tools, like various word embeddings, word vectorizors, and models. For the classification analysis, we attempted to utilize our own methods and methods provided by end-to-end text analytics libraries. In the latter case, we used the FastText library for a comparitive classifier.

Machine readible text representation Text is typically cleaned and, afterwards, converted to machine readable formats. Machine representation of natural language comes in many forms. There are simple procedures—like bag of words—or advanced models—like word embeddings—that can represent text in meaningful forms. Take the following sentences as an example to represent bag of words vectorization:

- Word vectors are simple
- Word embeddings are complicated

These two sentences could be converted to the following vectors—readible by computers. Table 1 shows the example of conversion to vectors. This representation of text is capable of being used inside a predictive model. However, there are downsides to the simplicity of the bag of words representation. One major issue is the context of each word is lost. We understand that certain words were mentioned, but there is really no sense of the word beyond being mentioned. In the table 1, the binary variable 1 represents if the word is present in that sentence, 0 otherwise. There are more sophisticated ways to represent the presence of a word, like based on frequency within the sentence iteself or the corpus of sentences, but these methods were avoided altogether. We found that other representations of text made significant improvements.

Word embeddings can be used for more sophisticated text representation. A word embedding is to be thought of as a large dictionary where large vectors of numbers are provided for each word in the dictionary. These large vectors, known as the embedding, are computed using machine learning on large bodies of text. The resulting embedding for a given word will convey its meaning. For example, the embeddings of the words "dog" and "cat" will be more similar to one another than the embedding for the word "shoe."

Common text cleansing As mentioned above, text is cleaned. Determining what is the best method of cleansing—or combinations of multiple methods—is a matter of trial and error. Machines may not find relevance in considerably equivalent words. For example, one method of cleansing is called stemming. After applying stemming to the following word, "swimming", the resulting stem would be "swimm." Notice the "mm" at the end of the word; the "mm" is not a typo. Stemming will, in many algorithms, simply drop endings like "-ly" or "-ing". Lemmatization is like stemming, but more complicated. Lemmatization will convert the raw word to the root of the word, or the lemma. For example, the word "swimming" will be reduced to "swim."

Validating the efficacy of models In machine learning, the main objective could be argued to maximize accuracy or minimize error on data never seen before. To do this, analysts will employ a training and testing sample. The training sample will consist of data that the model can use to learn the problem at hand. This means we give the model access to our independent and dependent variables. The testing sample, however, will not be given the dependent variables. Measuring the expected outcome versus the predicted outcome on data the model has not seen addresses the performance of the model.

Measures of efficacy The problems that we attempt to solve in this analysis are solely classification problems. Analysis throughout this project comes in two forms: binary and multiclass classification. Two main measures that are of interest in either classification problem are precision and recall. Precision is the ratio of true positives to total positive predictions. Consider the following confusion matrix in Figure **??**. The precision can be computed as

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{1}$$

which, when applied to the confusion matrix, becomes $\frac{100}{100+15} = 0.8696$. In this case, the model correctly predicted roughly 87% of positive predictions. Recall, the ratio of predicted true positives to actual true positives, can be computed as

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{2}$$

which, when applied to the confusion matrix, becomes $\frac{100}{100+5} = 0.9523$. In this case, the model correctly predicted roughly 95% of the positive case. This is also known as the true positive rate or sensitivity. Another relevant measure of performance is the false positive rate. This is the ratio of false positives to the true negative class. The false positive rate can be calculated as

$$FalsePositiveRate = \frac{FalsePositives}{FalsePositives + TrueNegatives} \tag{3}$$

which, when applied to the confusion matrix, becomes $\frac{15}{15+80} = 0.1579$. In this case, the model incorrectly labeled rougly 16% of observations as negative.

Within binary classification models, there exists a threshold where a prediction becomes a negative or positive case. In our models, the threshold is always constrained between zero and one. A prediction less than a threshold of 0.5, for example, becomes a negative case and, otherwise, a positive case. Therefore, there will always be an inherent tradeoff between a model's true positive rate and the false positive rate. Consider the following receiver operating characteristic curve in Figure 4.

The red line in Figure 4 represents a specific model's true positive rates and false positive rates at varying thresholds. The blue line represents a model with no predictive insight. This model is considered to be randomly assigning positive or negative classes. As you can see, it is in your interest to push the model's ROC curve up-and-left. Stated differently, maximizing the area under your ROC curve is a good measure of performance. This is known as area under the curve or AUC.

4.1 Binary Classifier

4.1.1 Method 1: Basic Models with Advanced Word Representation

Data modification Various iterations of data modification were tested. To gain an understanding of what could work best when modifying the data, we used a simple logistic regression to benchmark the impact of data modification.

Data was modified by two different combinations of methods:

- Remove stop words, convert words to lower case, and lemmatize words
- Remove stop words, convert words to lower case, lemmatize words, and remove digits

For clarity, stop words are commonly referred to words that do not add much meaning to a sentence. For example, consider words like the, he, have, or I. Removing these words remove unnecessary words and promote more succint sentence representation.

In the first method, we had the accuracy and error metrics are reported in Table 3.

Compared to the second method, the first method performed much better as seen in Table 4.

Unless otherwise stated, the first method will be maintained throughout the rest of the analysis.

Text representation Word embeddings were obtained from FastText's Wikipedia trained embedding. This word embedding represents text in a 1 by 300 vector. For each observation, word embeddings were obtained for the first 50 words. These words are filled first from the article's headline followed by the body of text if the headline is less than 50 words. If a specific word was not in the embedding's dictionary, that word is skipped. All word embeddings were then averaged. The resulting feature space for the model was a 1 by 300 vector.

Model results Logistic regression was used as a baseline model to understand performance capabilities. Logistic regression is a very simple model, yet its performance was quite good. In Figure 1, you can view the test set performance.
The ROC curve on the test set shows great results. The conclusion that the model has a great trade off with the training and testing samples can be justified by the table containing performance statistics. The fact that the in-sample and out-of-sample precision and recall scores are close indicates that further training would come at a cost of the test sample's performance. Refer to Table 5 for performance statistics.

4.1.2 Method 2: Advanced Models with Advanced Word Representation

Data modification The data for this method was modified in the exact way as the previous, best performing model. This was made to promote comparability with using more advanced training models.

Text representation Again, this model used the same representation of text as the previous, best performing model. For each news article, there was a 1 by 300 feature representation of select words from its headline and body.

Model specification The model tested in this analysis was a feed-forward neural network. This neural network used an input layer, a single hidden layer, and an output layer. For the input layer, there were 300 input nodes with rectified linear unit activation functions. This layer was fully connected to the hidden layer with 128 nodes with the same activation functions. These nodes connected to the final output node with a sigmoid activation function. There was a batch size of 10 during the 60 epochs of training.

Model results To maintain comparability between models, the same training and testing samples were used. After the training process, the model had the following performance statistics found in Table 6.

In the neural network performance table, we can see that this method was much more effective than the previous model.

4.1.3 Method 3: FastText Library

For additional analysis, we employed the FastText library for this classification problem. Compared to building your own procedures to classify text, this method is considerably more of a black box. However, this comes with added implementation simplicity at a cost of understanding the inner workings.

Model results After throwing in the raw text into this FastText engine, we achieve some good results. Refer to Table 7 for performance statistics.

Figure 5 shows the receiver operating characteristic curve for the test set. These results seem to have some degree of overfitting as the results in the training sample are considerably better compared to the testing sample.

4.2 Multiclass Classifier

4.2.1 Method 1: Logistic Regression

Data modification The data were modified using the same procedures outlined by the other models. This includes lemmatization and removing stop words.

Text representation As illustrated in other sections, the first 30 words from the headline and body are converted into a 1 by 300 vector. This feature space is then provided to the model for training and prediction.

Model results There are 16 separate classes in this classification problem. The training and testing performance tables can be viewed to compare precision and recall statistics. Refer to Table 8 for performance statistics.

The results for this multiclass classification problem are far from the performance of the binary classifier. Using logistic regression and this text representation method results in very poor results. Additional attempts to represent the text in a different fashion were tried. However, no method compared to the performance experienced by the FastText library.

4.2.2 Method 2: FastText

The FastText library was used for this classification problem. The raw text provided to this library and the following results were experienced. Refer to Table 9 for performance statistics.

4.3 Question and Answering

With the ability to classify general news articles from business to non-business, we could then classify the type of business news reported in the text. Afterwards, we want will want to extract relevant information from the articles.

One major area of interest was the mergers and acquisitions news category. Within these articles, major questions of interest were:

- Who was the buyer?
- Who was the seller?
- Who was the target?
- What was the acquisition amount?

To address this type of question and answering, we used a library called DeepPavlov—an open source conversational artifical intelligence framework.

4.3.1 DeepPavlov examples and usage

DeepPavlov is a very powerful framework. In our application, we were able to provide raw text as context and ask informal questions about this context. For example, in the context "Apple Inc. bought Tesla Inc. in 2019 for $50 billion. Elon Musk and Tim cook are happy!" we ask the following questions and obtain resulting answers in Table 10.

When you interact with this question and answering framework, you receive a confidence level and the predicted answer. In the example with the provided context, we can ask any question and receive an answer or a non-answer. This means that the framework can declare that the answer is not contained in the text.

4.3.2 Developing a ground truth dataset

In order to validate the efficacy of the DeepPavlov model, we needed to obtain a dataset with labels that were known as fact. Roughly 200 articles were classified with information relevant to mergers and acquisitions.

4.3.3 General analysis

Our main concerns for the mergers and acquisitions news type was discovering the following information:

- Who was the seller?
- Who was the buyer?
- What was the price of the acquisition?

DeepPavlov does have a degree of sensitivity to the wording of the question asked and the wording around the answer in the context. Therefore, it is essential to understand common words around the ground truth answers found in the text. This understanding was obtained through analyzing words leading up to the answer and words that followed the answer.

Before mention of the target In Figure 6, we can observe the frequency of words prior to the mention of the target. Within two words prior to the target, the most frequently used word was "of" followed by "acquisition." Most of the words following the buyer take on some form of the word "acquisition." When formulating questions, we will likely want to include a form of the word "acquisition."

After mention of the target In Figure 7, we observe the frequency of words after the mention of the target. The most common word was "to" followed by the word "from." A question like "Who was the company sold to?" could be effective due to these word frequencies.

Before mention of the seller In Figure 8, we observe the frequency of words before the mention of the seller. The most common word was "of" followed by the word "acquisition." Interestingly, these frequencies are very similar to those before the mention of the target. Therefore, in many cases, the target is also the seller.

After mention of the seller In Figure 9, we observe the frequency of words after the mention of the seller. The most common word was "to" followed by the word "a."

Before mention of the buyer In Figure 10, we observe the frequency of words before the mention of the buyer. In a significant amount of articles, the character "-" and the news vendor name "/PRNewswire/" was mentioned. This indicates that, in most articles, the buyer company name is found at the very beginning of the article.

After mention of the buyer In Figure 11, we observe the frequency of words after the mention of the buyer. There are not that many significant findings in the words after the buyer. One observation worth mentioning is that often company names are followed by their public ticker symbol in parenthesis.

4.3.4 Text preprocessing

Several iterations of text preprocessing was administered to see how performance changed in our ground truth dataset. We reduced variability in company names by replacing company names with "company" followed by an index. For example, the first company mention in a document would be replaced with "company0." We found that this method of complexity reduction was not helpful.

4.3.5 DeepPavlov performance

Comparing the performance of the DeepPavlov models is tricky as there can be variance in the information recorded in our ground truth dataset and what the DeepPavlov framework will yield in return. To account for this variance, the ground truth and the answer provided by the framework will be compared with a similarity metric, instead of full equality. In retrospect, our similarity measure may not have been entirely applicable as it was the en_core_web_lg-3.0.0 pipeline from the spaCy library. When comparing similarities between documents, we use a similarity threshold of 0.7 to determine if the documents are the same.

Who was the seller? For the DeepPavlov framework, multiple questions were asked to determine the seller. Of these multiple questions, the answer with the highest confidence score was selected as the result. Specifically, we asked:

- Who was bought?
- Who was acquired?
- Who was sold?

The variability in the formulation of the questions can address the variability of the context. DeepPavlov will find words of similar meanings to reach the answer. Better suited questions to the actual phrasing will promote better results. Out of 176 ground truth articles, 104 were predicted correctly using this method. This is roughly 59.09% correct. This performance could be improved greatly with a better ground truth data set and a better similarity measure.

Who was the buyer? Multiple questions were asked to determine the buyer for a news article. The phrasing of determining the buyer was of the following formats:

- Who was the buyer?
- Who acquired?
- Who acquired the company?
- Who is the buyer?

Asking multiple questions, and maximizing the confidence for a response results in much better performance. Out of 180 ground truth articles, 84 were predicted correctly—or 46.67%. As before, this performance could be improved through better tagging in the ground truth data set and more appropriate similarity measures.

What was the transaction amount? Multiple questions were asked to determine the transaction amount for a given transaction. These transaction amounts were specified with tremendous variability. In most articles, a transaction was not specified. When they were, the transaction amount was often not an explicit dollar figure. Again, a series of questions were used to maximize confidence:

- How much?
- How much was it sold for?
- What was the transaction amount?

Out of 25 articles that included a transaction amount, the framework was able to correctly predict 15 transaction amounts. This was roughly 60% of the data set. There was no testing over the entire ground truth data set. Therefore, this 60% accuracy could be thought solely as a true positive rate. The framework can predict a non-answer, so the accuracy over the entire data set could be better.

4.4 Company tagging

Obtaining information about companies is only useful if these insights can be stored and tied back to company profiles. To establish this tagging, we needed to create an index for all company mentions for a central repository.

4.4.1 Identifying company names

From the spaCy library, we were able to use their part-of-speech tagging framework. This enabled us to extract parts of speech like locations, organizations, people, and dates from raw text. In Figure 12, we can observe spaCy's parts of speech tagging framework. We are interested, primarily, the organization tag.

For a given company, its name in text can vary from truncated forms compared to its legal name. These subtle variations can be addressed with similarity measures. For example, in Figure 12, the organization "HigherGround Managed Services" is also referred to as "HigherGround." These two names represent the same company but are written differently. This is addressed by measuring similarity between organization names. If two organization names have a similarity above our threshold at 0.7, they can effectively be considered the same organizaiton name. Storing these names and their variations allows us to develop an index to match insights from news articles.

4.5 Implementation

Going forward, all news articles should be pass through a part of speech engine to extract organizations and assigned an index. These indices should include similar names for future pairing to insights from news articles.

5 Recommendations and Operational Execution

5.1 Value creation

Primarily, we have demonstrated a proof of concept for future analysis and implementation. We have shown what is possible and the levels of accuracy to be expected. There can be many improvements, but we have paved a way for future analysis. Future analysis could be based on the ground truth dataset we created for the insight extraction phase of this project.

5.2 Next steps

The next steps would be to create a pipeline to go from raw text to a structured database with new insights extracted from text. These pipelines should be stress tested to see how many articles could be processed per day. Additionally, further steps should be taken to build the types of questions asked to articles. There are many other news types that have yet to be utilized. There is more value to be uncovered in the world of unstructured data.

6 Conclusions

We have been able to predict the business nature of raw news articles. This includes determining if, broadly speaking, the content relates to business. Further, these raw documents could be classified according to the subject of the business news. After the news has been classified according to its subject, we determine a series of facts potentially expressed in the mergers and acquisitions news type. Next, we create methods to match these insights to a structured database containing other company information. In its entirety, we are able to create value, in the form of company data, from unstructured data. Our methods alleviate a significant amount of effort from an equivalent, monotonous task done by human intervention.

7 References

This analysis made many references to the tutorials on the libraries we used for our analysis. The main references we made were to three main libraries: DeepPavlov, NLTK, and spaCy.

- spaCy: https://spacy.io/
- Natural Language Tool Kit (NLTK): https://www.nltk.org/
- DeepPavlov: https://deeppavlov.ai/

8 Acknowledgments

This analysis would not have been as educational if it was not for our project supervisors at Dun and Bradstreet. Their dedication to our project was the reason this was a valuable learning experience.

9 Figures

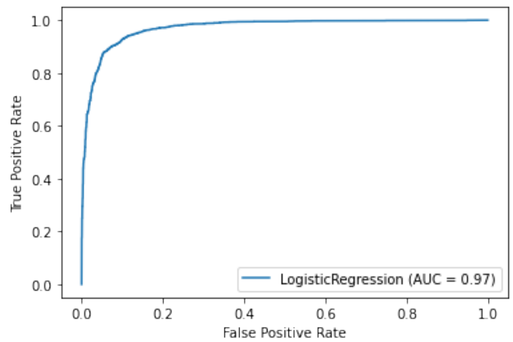

Figure 1: Test set performance with logistic regression

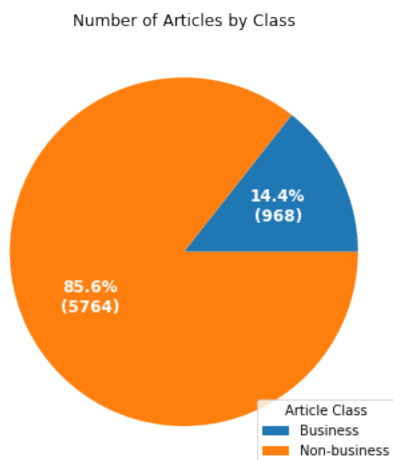

Figure 2: Distribution of article classes in dataset

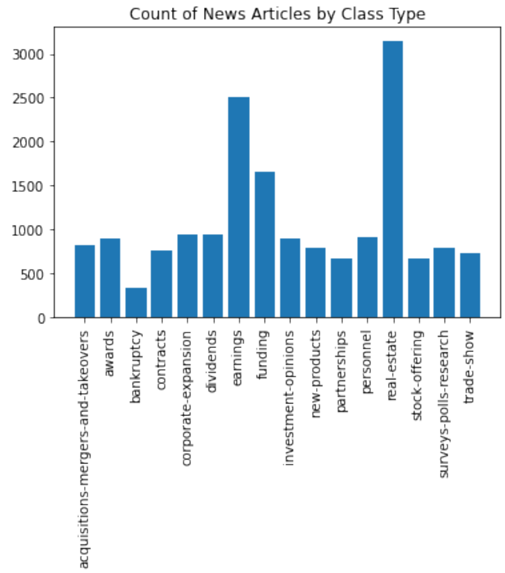

Figure 3: Distribution of article classes in dataset

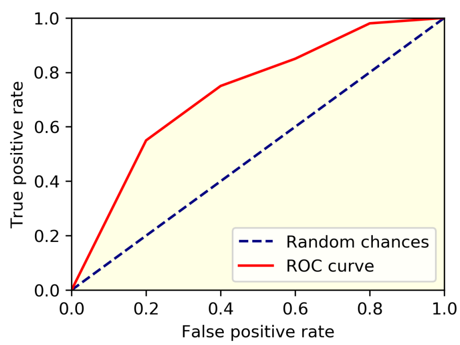

Figure 4: Receiver operating characteristic (ROC) curve

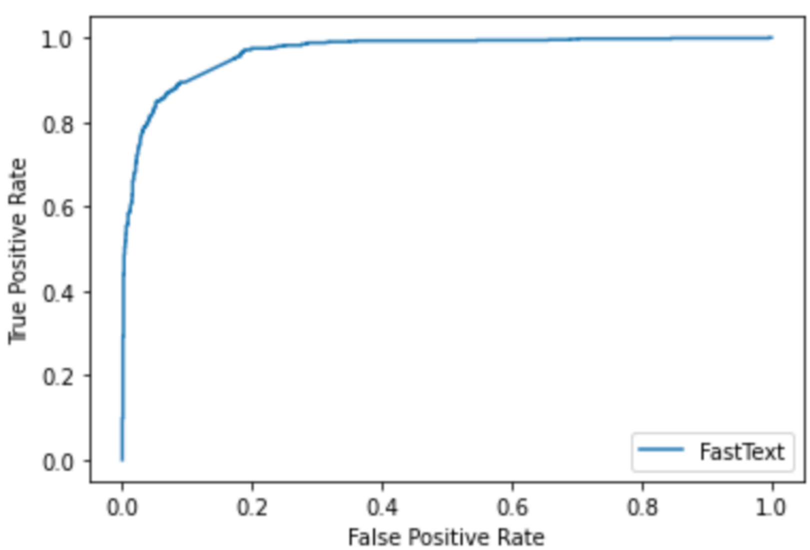

Figure 5: Test set performance with FastText

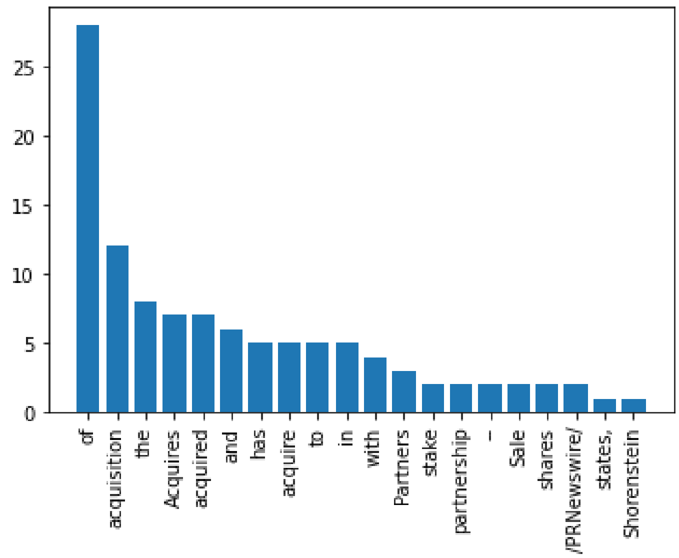

Figure 6: Frequent words prior to the target

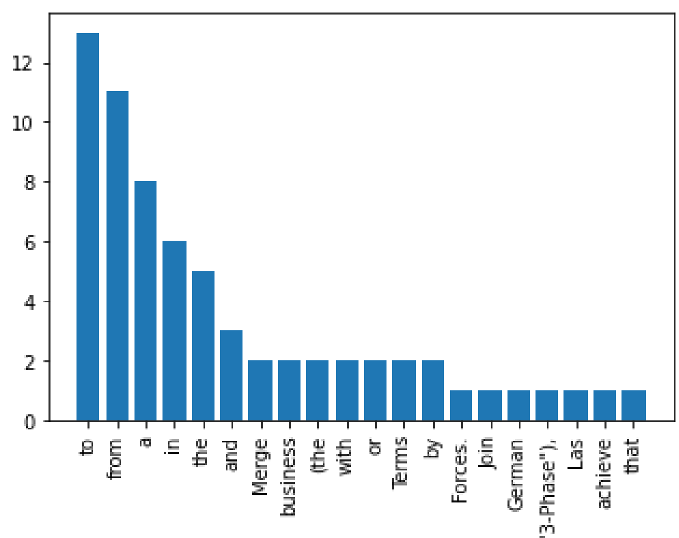

Figure 7: Frequent words after the target

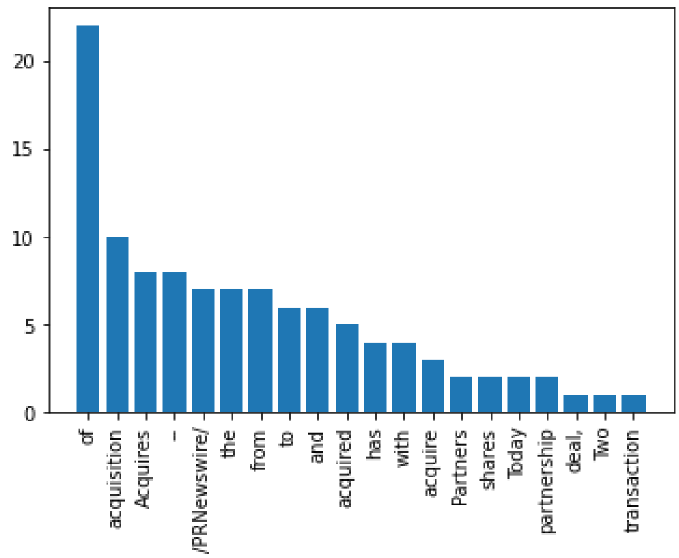

Figure 8: Frequent words prior to the seller

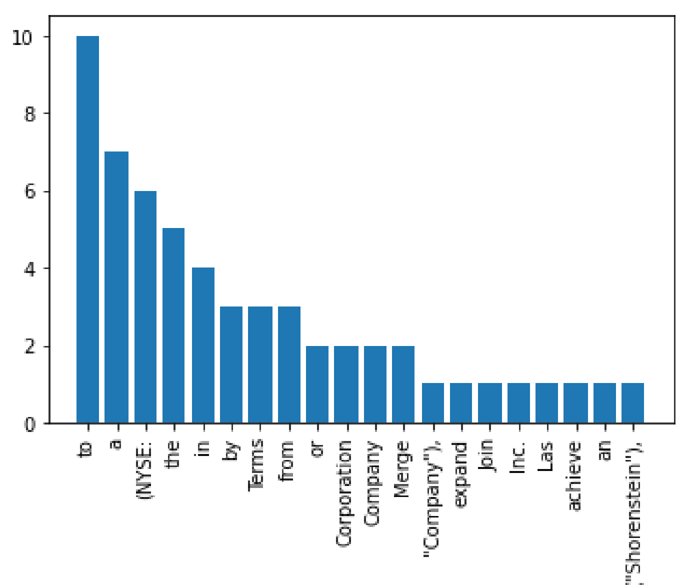

Figure 9: Frequent words after the seller

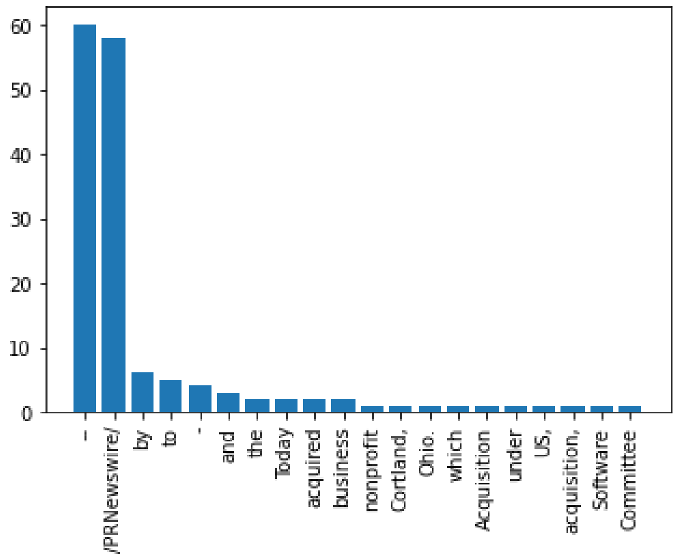

Figure 10: Frequent words prior to the buyer

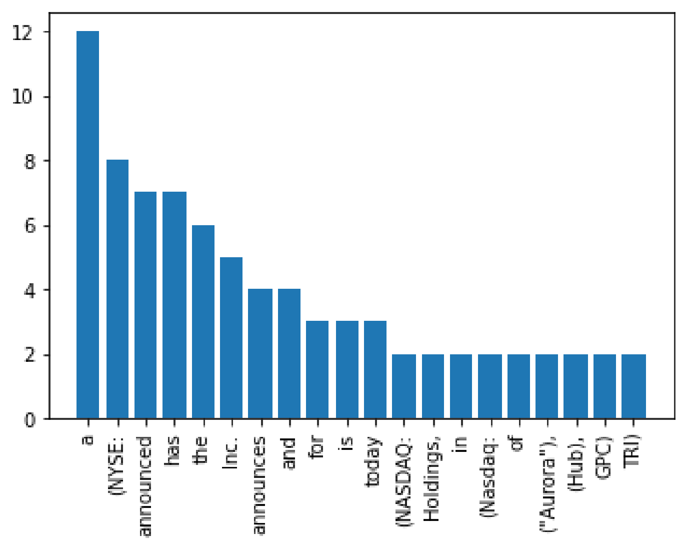

Figure 11: Frequent words after the buyer

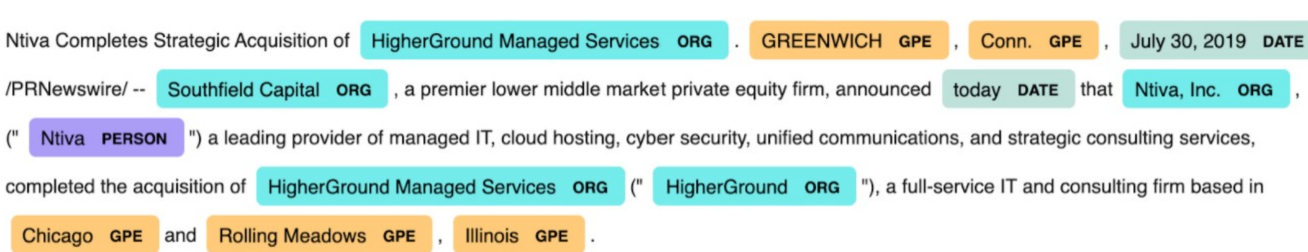

Figure 12: Parts of speech tagging example

10 Tables

Table 1: Visualization of Bag of Word Vectors

	word	**vectors**	**are**	**simple**	**embeddings**	**complicated**
Sentence 1	1	1	1	1	0	0
Sentence 2	1	0	1	0	1	1

Table 2: Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	100	5
	Negative	15	80

Table 3: Preprocessing: Version 1

Metric	**Test**	**Train**
Precision	0.889039	0.894007
Recall	0.921459	0.905754
F-Score	0.904959	0.899842

Table 4: Preprocessing: Version 2

Metric	**Test**	**Train**
Precision	0.886054	0.900621
Recall	0.730715	0.722700
F-Score	0.800922	0.801911

Table 5: Logistic regression performance

Metric	**Test**	**Train**
Precision	0.891304	0.895208
Recall	0.920056	0.905754
F-Score	0.905452	0.900450

Table 6: Neural network performance

Metric	**Test**	**Train**
Precision	0.891304	0.932077
Recall	0.941094	0.963752
F-Score	0.918549	0.947650

Table 7: FastText performance

Metric	**Test**	**Train**
Precision	0.9113	0.9752
Recall	0.8918	0.9329
F-Score	0.9014	0.9536

Table 8: Logistic regression performance on test sample

Class	**Precision**	**Recall**
acquisitions	0.65	0.63
awards	0.74	0.73
bankruptcy	0.86	0.55
contracts	0.49	0.40
corporate-expansion	0.58	0.45
dividends	0.80	0.75
earnings	0.80	0.94
funding	0.71	0.72
investment-opinions	0.86	0.90
new-products	0.55	0.56
partnerships	0.48	0.27
personnel	0.74	0.75
real-estate	0.70	0.74
stock-offering	0.74	0.83
surveys-polls-research	0.60	0.72
trade-show	0.63	0.56
macro avg	0.68	0.66
weighted avg	0.70	0.71

Table 9: FastText regression performance on test sample

Class	**Precision**	**Recall**
acquisitions	0.8860	0.7149
awards	0.9260	0.7796
bankruptcy	0.86	0.9155
contracts	0.8923	0.5759
corporate-expansion	0.8851	0.6345
dividends	0.921	0.8434
earnings	0.9321	0.9107
funding	0.9283	0.7553
investment-opinions	0.995	0.9689
new-products	0.8868	0.6125
partnerships	0.9318	0.5989
personnel	0.9125	0.7716
real-estate	0.8851	0.7617
stock-offering	0.9141	0.8663
surveys-polls-research	0.9665	0.8243
trade-show	0.8699	0.678

Table 10: Questions posted to DeepPavlov using provided context

Question	**Answer**	**Confidence**
Who was bought?	Tesla Inc.	1878
When was the company bought?	2019	106235
How much was the company acquired for?	$50 billion	7883
Who was the buyer?	Apple Inc.	4