**Task 4: Hyperpartisan news detection technical report**

**1. Problem presentation**

News are our portal to the events happening in the world. As consumers of news, we have to distinguish between facts and fabrications, between objective truths and opinionated stances, between indisputable reality and unreliable falsehood. Everyday, we are prone to being influenced or manipulated by the more opinionated news or by those that align with our own thoughts and views of the world, confirming our biases and strengthening our prejudices. Thus, we need reliable means of telling whether an article is merely a fact-stating objective piece of news or if it has someone else’s interest on its mind.

**2. State-of-the-art**

Our specific subject, “Hyperpartisan News Detection”, is, as of right now, mostly unexplored territory. Even though the amount of Fake News detection algorithms boomed in the last couple of years, programmers and linguists alike are still dipping their toes in the more subtle domain of written opinion alignment.

We expect more solutions to arise as the SemEval competition draws closer to its conclusion. We are certain the competition motivated many teams to tackle the elusive subject of written text classification, and are eager to compare and share our results and findings with others. The chance of being awarded is less enticing than the thought of having created something that matters, and further developing approaches and solutions for this sprawling domain into its ever-growing grip of relevance in the way we consume news.

**3. Solution**

Our solution consists of 2 different heuristics which compute a score for any given article, and, based on certain filtering criteria, estimates whether the article follows a hyperpartisan type of argumentation. These heuristics are: word frequency and sentiment analysis. Stance detection was planned as well, but was ultimately discarded.

3.1. Word frequency  
 Using the training dataset, this heuristic computes the percentage that every single relevant word appearing in a hyperpartisan-type argumentation has. In order to do this, first, we create a dictionary with the words themselves as keys and their respective frequencies (total number of appearances) as the values. The criteria for a new word to be accepted into the dictionary is to not be a stopword (filtering); this way, we obtain only words with at least minor relevance. Afterwards, we count every word from this dictionary in hyperpartisan-type of argumentations in the dataset. At the end of this process, we will compute the percentage any word has to be in a hyperpartisan-type of argumentation.  
 It stands within reason that some words - outliers with percentage scores over 80%, for example - prove vital in correctly classifying some articles, but the number of such words is relatively low.  
  
3.2. Sentiment analysis  
 Due to the shortcomings of the previous heuristic, this calculation was made with the aim of classifying all articles based on the emotional neutrality of the author. In other words, this heuristic searches for “positive words” and “negative words” based on the SentiWordNet scoring and simply computes the total score of every single (relevant) word in any article. We have considered that should an article be mostly or totally positive or negative, then, said article does not present both sides of the story and are, therefore, hyperpartisan articles.  
While this heuristic could, theoretically classify every article correctly, the usage of sarcasm would get wrong scores for articles, and computing the disambiguation for vague words or expressions would take too much time, giving too little increase in accuracy.

3.3. Stance detection  
 Stance detection’s original purpose was figuring out whether an article is outlandishly partisan - or downright fanatical - in its argumentation. Because of unforeseen circumstances related to its implementation, the results clashed constantly with those we obtained using sentiment analysis, so we ultimately decided to scrap this one altogether.

**4. Results & Evaluation**

The end result was obtained by combining both heuristics and the extensive training data provided by the competition organizers in a single evaluation process that used the cross-validation method for assessing accuracy.

The impact of both heuristics was tweaked around 200 times in an autonomous process, serving the purpose of finding a balance between word-counting and sentiment analysis. Different pairs of ‘weights’ (adding up to 1) were tried and their results registered. The balance struck somewhere around 62% for word counting and 38% for sentiment analysis - basically, word score counting did the blunt of the work, whereas sentiment analysis helped smooth the way and ever-so-slightly tip the scales in cases of uncertainty.

The cross-validation accuracy ended up being in the range of 85-86%. Altough it definitely could have been improved, we feel that this is a decent result and should prove useful (and accurate) in most cases. It is, by no means, a benchmark for other results, and we see it strictly as something to improve upon.

On the other hand, achieving close to 100% accuracy would not have been desirable either. The competition organizers warned participants that the training data and the evaluation data will have no authors in common - that means that certain patterns of speech might occur more often or not at all when compared to the data we trained our algorithm on - in other words, we want to avoid tailoring the program to our specific data set *too much*, and thus avoid overfitting. We think a result somewhere in the 90-95% range would have been most appropriate for the challenge at hand.

**5. Comparison with other solutions**

There isn’t much to add here yet. Once the SemEval results are published, we’ll have plenty of other solutions to compare our results to.  
  
**6. Future work**

Everything can be improved, and this project is no exception. Besides fine-tuning the weights even further and other small tweaks to formulas and so on, our solution could also benefit from approaches and strategies we may not have considered. More often than not, seeing different solutions helps the creative process rather than hinder it, and we believe this to be the case here as well. Seeing how others tackled the problem might motivate us further, and we would most definitely like to re-pick this up as a pet project in the future.

We believe reaching 90% accuracy is entirely possible and would definitely benefit from seeing new ideeas applied by our peers.

**7. Conclusion**

Hyperpartisan News Detection was, for us, equally challenging and fun. It made us think outside the box and consider things we haven’t entertained in thought before, and look for solutions outside the realm of definitive science and more into applied theories, so to say. We had to come up with our own plans and executions rather than follow the well-known approach and engineer new ways of tackling previously-unknown challenges. The fact that our end result was merely decent, rather than particularly remarkable did not dishearten us, but motivated us further and we eagerly await the new step of the competition in order to properly assess our performance and compare our standings.

**8. Bibliography and Links**

· http://deeplearning.net/  
· https://eu.udacity.com/course/natural-language-processing-nanodegree--nd892  
· ttps://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e  
· https://deeplearning4j.org/  
· https://chainer.org/  
· https://keras.io/  
· https://stanfordnlp.github.io/CoreNLP/  
· https://web.stanford.edu/class/cs224n/reports/2754942.pdf  
· https://arxiv.org/ftp/arxiv/papers/1701/1701.00504.pdf  
· https://www.nltk.org/