Automatic Extractive Summarization for NewsWire Data

Ayushi Aggarwal
University of Washington
ayushiag@uw.edu

Elijah Rippeth
University of Washington
rippeth@uw.edu

Seraphina Goldfarb-Tarrant
University of Washington
serif@uw.edu

Abstract

We developed an end-to-end extractive text summarization system for the Text Analysis Conference[1] shared task. We utilized LexRank (Erkan and Radev, 2004) to compute sentence salience scores, supplemented with various embedding techniques, including TF-IDF, GloVe vectors, and Doc2Vec vectors. Additionally, a chronological content ordering expert adds to the general coherence of candidate sentences. We found that the TF-IDF embeddings outperformed the next most competitive embedding measure by as much as 37.6% in F1 in ROUGE scores, though other embeddings were judged to be qualitatively superior.

1 Introduction

The Text Analysis Conference (TAC)[1] summarization task provides a large collection of newswire articles to be compressed into short (100 words) summaries by topic. Each topic is given, and contains ten documents from disparate news sources, all in English. As per the task, the summaries are evaluated for content and responsiveness. With this goal in mind, we approach automatic summarization of the given dataset. The two main approaches by which text summarization is realized are abstractive, which focuses on capturing concepts and generating potentially novel utterances(paraphrases), and extractive, which filters out the least salient sentences from a corpus, leaving only the most relevant information.

We attempt to develop an extractive summarization system. In this paper, we explore the various baselines for content selection and ordering, develop a working end-to-end baseline and evaluate the summary outputs for quality by looking at coherence, informativeness and anaphora resolution. We also present the results of ROUGE score analysis and hyper-parameter tuning as detailed in the Results and Discussion sections.

1.1 System Architecture

The main modules of the architecture are enumerated below with details of third-party packages and APIs they utilize. Details of the functionality are provided in the Approach section.

1. **Corpus Reader:** This module is tasked with reading in the dataset as provided for the TAC shared task. It utilizes lxml[2] and BeautifulSoup[3] for XML parsing and passes the resultant to the preprocessor module. PyYAML was used for custom end-to-end system configuration.

2. **Preprocessor:** The preprocessor resides within the corpus reader as as a method and stores all documents for a Topic as Story data structures that further have headings and Sentence objects. Sentence objects store metadata about the sentence (order in original document, date of original document) and configurable SentenceEmbedding objects. For each topic, we experiment with tokenization via using either the NLTK[4] Punkt Sentence Tokenizer (Kiss and Strunk, 2004) and TreeBank Word Tokenizer, or SpaCy's[5] default and dependency parse tokenizers. All tokenizers available report high reliability across Precision, Recall, and F1, but given that LexRank operates with a sentence as a basic unit, and that it operates on relatively small

[1] https://tac.nist.gov/tracks/index.html

[2] http://lxml.de/

[3] https://www.crummy.com/software/BeautifulSoup/

[4] https://www.nltk.org/

[5] https://spacy.io/usage/linguistic-features

numbers of sentences, even small errors in tokenization have the potential to significantly effect performance, which we found to be the case empirically in our initial observations of the system.

3. **Configurable summarizer:** This is the main module in our pipeline, which consumes a Topic and outputs a ranked list of candidate sentences. It does further pre-processing if necessary for a given SentenceEmbedding strategy. When stemming is performed we use the NLTK Porter Stemmer [6] (Porter, 1980). The summarizer utilizes a customized fork of the sumy[7] implementation for LexRank technique for computing sentence saliency scores. This is further detailed upon in the section that follows. Numpy was utilized for linear algebra calculations as a part of LexRank.

4. **Post-processing scheme:** This scheme generates the final summary as expected by the TAC task definition.

5. **Information orderer:** The information ordering module orders a set of candidate sentences which are the output of the summarizer according to various metadata. This is all original code which relies on extension of an Expert interface for various ordering conditions. This will be discussed in detail in a later section.

The overall system architecture is illustrated in the figure 1:

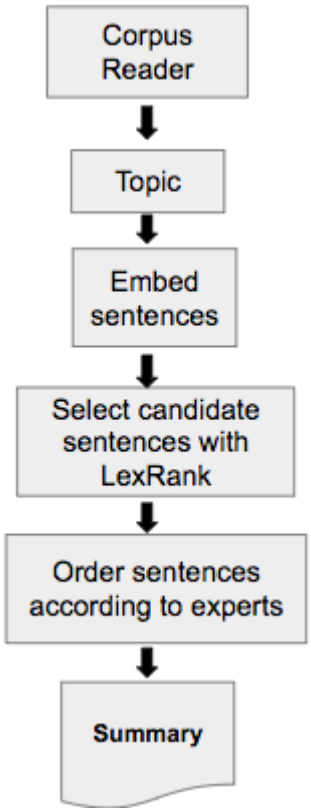

Figure 1: The system summarizing a single topic with an arbitrary sentence representation.

2 Approach

This section details the functionality of the core components of the summarization pipeline.

2.1 Dataset and Preprocessing

In the interest of making the system as flexible as possible for different experimental setups, and for the ease of parallel project group work, we operated on configuration files from the onset. The TAC dataset is categorized by topic, with each topic consisting of a set of documents. Since the task is topic-orientated and the ordering of documents does not matter, all documents are combined into one giant document for each topic. The **corpus reader** stores the pointers to the relevant documents in the corpus grouped by topic set. This allows us to avoid the potentially expensive task of storing entire documents and operate solely on a subset of the topics covered by the dataset.

The **corpus reader** proved to be the most tedious and time-consuming module to build in the baseline pipeline. Inconsistencies with formatting of documents in different datasets required separate ingest mechanisms, which proved to be fairly error-prone and therefore, a time drain.

2.2 Content Selection

The system takes the sentences for each Topic as the input to the **summarize** module. Because of the structured nature of the domain newswire articles, where the beginnings of documents often are natural summaries, we implemented a quick-and-dirty baseline that chooses the first sentence from each Story in the Topic and generates a summary. We used this as our base metric upon which to improve. We also implemented a baseline that selects

[6] http://www.nltk.org/howto/stem.html
[7] https://github.com/miso-belica/sumy/tree/dev/sumy/summarizers

sentences randomly from each document, to establish a performance lower bound. The final system implements a graph based sentence saliency score computation method called LexRank (Erkan and Radev, 2004), which is a modification of the famous PageRank algorithm (Brin and Page, 1998). All sentences in a topic act as the nodes of the graph. Weights for the edges are calculated by considering sentences as bag-of-words and computing the TF-IDF cosine similarity between sentences and centrality score for each sentence node *v* computed as the fraction of the sum of the centrality scores of all the neighboring nodes to the degree of the node. A matrix of the LexRank scores is calculated using the centrality score for each sentence, from which we can select the top *n*. Once of the initial reasons we chose LexRank was its ease of adaptation to different experiments. Abstractly, LexRank operates based on sentence similarity; the LexRank framework can be used on any set of sentences, given some way of calculating a similarity score between them. We experimented with substituting the TF-IDF representations of sentences with other options. We tried averaged GloVe embeddings (Pennington, 2014), provided by spaCy[8], which are trained on the Common Crawl Corpus.[9]. We also tried Doc2Vec (Le and Mikolov, 2014), trained via Gensim[10] on the English Gigaword Corpus [11]. We built the **summarize** module to be easily extensible to any new models of representing sentences, should it make sense to try more in future.

2.3 Information Ordering

Information adding was added as a post-processing measure in hopes to improve coherence between sentences from disparate stories. To this end, we leverage a weighted linear sum of expert systems, first proposed in "A Preference Learning Approach to Sentence Ordering for Multi-document Summarization" (Bollegala et al., 2012), to determine whether sentence u or v should be preferred given a partial summary Q:

$$p(u, v, Q) = \sum_{e \in E} w_e \cdot q(e, u, v, Q)$$

where E is the set of experts, w_e is the contribution a given expert makes to a preference, and q is a function with a domain [0, 1] that defines the amount of preference to be given to sentence e.

By defining ϵ as the threshold above which to prefer u, we can order all sentences in a document by iterating over the unique pairs (u, v), of combinations of all sentences and voting for u if $p(u, v, Q) > \epsilon$ and v otherwise. The order is determined by a reverse sort of votes.

To study how ordering helps improve the system, we defined a single information ordering expert based on (Bollegala et al., 2012) for chronological ordering. This method relies on metadata from the articles from which the sentences are extracted; specifically, it relies on document id ($D(x)$ is defined to be a function returning the document id of sentence x), publication date ($T(x)$ is defined to be a function returning the date the document containing sentence x was published), and the position of the sentence in a document ($N(x)$ is defined to be a function returning the index of a sentence in a document).

Ordering of the chronological expert is as follows:

$$q(c, u, v, \_) = \begin{cases} 1 & T(u) < T(v) \\ 1 & [D(u) = D(v)] \wedge [N(u) < N(v)] \\ 0.5 & [T(u) = T(v)] \wedge [D(u) \neq D(v)] \\ 0 & otherwise \end{cases}$$

In other words, if a sentence u was published before sentence v, u should be preferred earlier in a summary. Similarly, if two sentences are from the same document, the earlier sentence in the document should be preferred. If two unique documents were published on the same date, no preference should be given. Otherwise, prefer v.

With this single, simple heuristic, we noticed improved qualitative coherence in our summaries. We hypothesize that other experts may improve coherence additionally.

2.4 Post-processing

The ordered output obtained from LexRank is further processed via truncation at the sentence level to adhere to the final summary word limit of 100. We include the highest scoring first *n* sentences that fall within the word-limit. The user can modify the summary word-limit for each topic by resetting the hyper-parameter *word_num*.

[8] https://spacy.io/models/en#en_vectors_web_lg

[9] http://commoncrawl.org/the-data/

[10] https://radimrehurek.com/gensim/models/doc2vec.html

[11] https://catalog.ldc.upenn.edu/ldc2011t07

3 Evaluation Results

3.1 Initial Round of Experiments for Baseline

Tables 1, 2 and 3 depict the results for experiments run on the DevTest data given for TAC for the random, first-sentences, and LexRank systems. Henceforth, unless otherwise stated, LexRank scores refer to LexRank with it's best-performing configuration: TF-IDF, with a threshold of 0.1. ROUGE scores are reported as the mean of multiple runs, as neither resolution of ties for LexRank nor post-processing truncation is deterministic, so different runs of the system will display small, non-significant differences in results.

Overall, it was surprising to see such low ROUGE-1 scores for all of the approaches. Our ROUGE scores were well below those reported in the LexRank paper (Erkan and Radev, 2004). In contrast, the first-sentences approach outperformed the other two baselines in every category apart from ROUGE-1 Precision, wherein LexRank narrowly beat the first-sentences approach.
Table 4 shows how first-sentences and LexRank have very high ROUGE scores with respect to each other, even though they are qualitatively quite different. This is further discussed in the following section. The results show the impressive level of unigram and even 4-gram overlap between the results of the two systems. For the given corpus, selecting the most connected sentences often ends up selecting the first one, though no sentence positional information is taken into account.

The effect of tuning the threshold hyperparameter for LexRank edge weights was also studied. As shown in Table 5, the best ROUGE-1 scores are obtained for threshold value of 0.1.

3.2 Sentence Embedding Experiments

We anticipated some weaknesses of TF-IDF - namely, that it should overvalue rare words, and that it captures only relatively local information as it treats sentences as unit vectors weighted by their TF-IDF scores within a sentence and document cluster. To ameliorate these, we extended our system to utilize GloVe and Doc2Vec embeddings. The results are summarized in Table 6.

We chose to use spaCy's GloVe embeddings because of their ability to capture global collocations, as the Common Crawl corpus is designed to be a representative sample of text on the internet. We also tried Doc2Vec embeddings, because of results from (Baroni, 2014) that show that

Type	P	R	F1
ROUGE-1	20.39	16.77	18.28
ROUGE-2	4.37	3.69	3.98
ROUGE-3	1.35	1.18	1.25
ROUGE-4	0.59	0.52	0.55

Table 1: ROUGE scores for random-sentence baseline run on DevTest data. (mean of 10 runs).

Type	P	R	F1
ROUGE-1	26.91	23.08	24.68
ROUGE-2	8.13	6.99	7.46
ROUGE-3	2.65	2.32	2.46
ROUGE-4	1.17	1.04	1.10

Table 2: ROUGE scores for first-sentence baseline run on DevTest data

Type	P	R	F1
ROUGE-1	28.95	21.98	24.75
ROUGE-2	7.70	5.82	6.57
ROUGE-3	2.49	1.91	2.14
ROUGE-4	0.81	0.64	0.71

Table 3: ROUGE scores for LexRank run on DevTest data

Type	P	R	F1
ROUGE-1	52.49	51.88	51.71
ROUGE-2	41.13	40.16	40.30
ROUGE-3	36.69	35.58	35.82
ROUGE-4	33.87	32.77	33.01

Table 4: ROUGE scores between LexRank and first-sentences, on DevTest data

Threshold	P	R	F1
0.1	29.239	21.805	24.693
0.2	28.185	22.115	24.55
0.4	22.616	18.057	19.99
0.5	21.673	16.879	18.877
0.9	21.286	16.572	18.555

Table 5: ROUGE-1 scores for different thresholds for LexRank

predictive models perform better than count models. Baroni's results are somewhat contradicted by (Levy, 2015), but Levy's work nonetheless recommends predictive models as the most reliable all-around solution as it does not perform poorly on any specific tasks. We expected Doc2Vec to also benefit from being trained on a matching domain, as the Gigaword corpus is a superset of the TAC dataset. Finally, Doc2Vec avoids the problem of averaging vectors to represent a sentence from its component words, though it is admittedly unclear exactly what information is being captured by training CBOW word embeddings in combination with Paragraph IDs. (Le and Mikolov, 2014). Still, as a result of these factors, we expected Doc2Vec to yield the best performance. Our experimental setup also included removal vs. non-removal of stop-words (using spaCy's stop-word list, which is 307 words long and of unknown provenance, but is slightly less conservative than NLTK's stop-word list, which is based off (Porter, 1980)). TF-IDF naturally downweights stop-words, but other methods of embedding do not, and we hypothesized that stop-words might muddy sentence embeddings, especially GloVe embeddings, since they are averaged. Thus, we expected stop-word removal to improve performance. All embeddings are 300-dimensional.
However, as is clear from Table 6, TF-IDF beats all other representations by a very wide margin for every ROUGE measure, and stop-word removal has a negative effect on ROUGE scores for alternate embeddings. TF-IDF does not clearly beat all other representations qualitatively, as will be discussed in further sections, but its performance does suggest that further work may be well spent by optimizing the system on top of TF-IDF rather than substituting different representations for computing sentence similarity.
Surprisingly, stop-word removal led to a decline in ROUGE scores for both GloVe and Doc2Vec. The reason for this is unclear.

4 Discussion

4.1 First-sentences vs. LexRank as Baseline Content Selection

As expected for the inverted pyramid structure of Newswire corpus content, the first-sentences baseline is hard to beat. It was in fact, *harder* to beat than anticipated, as LexRank did not outperform first-sentences in any area save ROUGE-1 Precision, in which the difference is not large enough to be reliably repeatable.

However, the results were qualitatively different. Many first sentences are a type of short abstract summarizing all topics that follow, but some are specific appeals to emotion or descriptions of an incident that were meant to introduce the feel of the narrative. For example, in summaries about the repeal of a pharmaceutical drug, some first sentences look like, *'When Emily Martin was hospitalized for emergency gallbladder surgery last summer, her doctors found that she had also had acid reflux, causing erosion of her esophagus.'* Where *Emily Martin* is not actually important and does not appear later, or in the model summaries; *she* is there to provide a human element to a broader story. In these cases, another standard narrative convention works *against* the first-sentences heuristic, just as the pyramidal structure works towards it. LexRank avoids this problem by valuing connectedness; these sentences are very detailed, and positionally important, but not very connected.

The downside of the LexRank structure is the product of its components. Cosine similarity between sentences will bias towards longer ones, and TF-IDF will overweight rare words. Our experiments display this tendency. We further analyzed the sentence length and lexical diversity (defined as the ratio of unique to total words, per document) of the model summaries and the LexRank baseline summaries by performing a Mann-Whitney U test[12] on the distributions over the two summary collections. We pre-selected $p = 0.01$ and found that the difference between both distributions easily passed the significance test. The average number of words and Lexical diversity Ratio are presented in Table 7. Since the model summaries are drawn from seven separate human summarizers, and are still so consistently shorter in sentence lengths and poorer in lexical diversity, it may be rewarding to look into ways to better approximate the style of human summarizers. That said, such a change would come with costs - for instance, shorter sentences will require more anaphora and co-reference resolution.

[12] https://en.wikipedia.org/wiki/Mann\%E2\%80\%93Whitney_U_test

	ROUGE-1			ROUGE-2			ROUGE-3			ROUGE-4		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
TF-IDF	**28.55**	**21.33**	**24.15**	**7.55**	**5.63**	**6.38**	**2.43**	**1.81**	**2.05**	**0.86**	**0.65**	**0.73**
GloVe	20.83	16.60	18.38	4.48	3.60	3.98	1.55	1.25	1.38	0.57	0.46	0.50
GloVe (no stops)	19.68	12.02	14.74	4.51	2.79	3.41	1.27	0.78	0.95	0.34	0.22	0.26
Doc2Vec	20.78	16.52	18.28	3.85	3.13	3.43	0.93	0.78	0.84	0.25	0.23	0.24
Doc2Vec (no stops)	20.58	10.48	13.61	3.42	1.78	2.30	0.95	0.52	0.66	0.24	0.14	0.18

Table 6: ROUGE results with LexRank for different sentence embeddings.

Type	**Avg Words**	**LDR**
Model	20.19	0.69
LexRank	28.25	0.73

Table 7: Distributional Statistics - Average number of words(**Avg Words**) per sentence and the Lexical Diversity Ratio(**LDR**) for the Model and LexRank summaries.

4.2 Sentence Embeddings

Our experiments with alternate sentence embeddings trained on larger corpora were much worse than we expected, especially in light of belief that data abundance leads to better trained systems[13]. Unfortunately, it is actually hard to analyze *why* they performed so poorly. GloVe should capture more global context, but perhaps it is capturing the wrong global context from a very general training corpus, and the more hyper-local awareness of TF-IDF within a document cluster is superior. That could also be a reason for Doc2Vec's weak performance, as neither represents the specific context of the cluster being summarized. An area for future work could be to control variables to be better able to compare the different embeddings. That said, (Levy, 2015) experimented with nine hyperparameter settings for five types of embeddings, concluding that the individual choice of model matters less than the hyperparameter tuning. In this light, it seems prudent to optimize for one embedding type.

Despite the weak ROUGE scores, we did notice qualitative improvements in Doc2Vec, discussed below. The ability to extend LexRank to new sentence representations is also valuable for further research and may be necessary for representing smaller chunks of sentences for redundancy reduction.

[13]This is broadly true across many tasks, and is one of the purported benefits of the GloVe vectors of (Pennington, 2014), which are faster to train and would therefore get performance boosts from consuming additional data

While Chronological sentence ordering adds some coherence to the discourse, we observe that SpaCy GloVe vector based system generates a summary wherein an incomplete sentence features as the second sentence in the summary. Doc2Vec based systems add to the summary quality over Spacy's GloVe based systems which tend to grant higher salience to quoted sentences and phrases. Doc2Vec summaries often end with a concluding statement, which is positive, but take a leap in discourse, therefore breaking the continuity of the summary. These examples (Section 6) highlight the major weaknesses of the current system, namely some extent of discourse incoherence, severe lack of anaphora and coreference resolution. A possible solution to these problems could be to make use of discourse and semantic features. It is worth noting that incorporating discourse features would require a mostly supervised system. This could prove to be tedious and non-feasible since human annotated discourse features are not scalable for a large dataset. Furthermore, the feasibility and pay-off of incorporating semantic features remains to be explored given that typically semantic features are not very helpful towards selecting summaries.

It was also noted that the summaries generally contain longer sentences. This could be tackled by normalizing the LexRank sentences saliency scores for sentence length to include more informative shorter sentences. Methods of event extraction to ensure only one sentence per event, or of distributional semantics to restrict synonymy will certainly improve performance qualitatively, as that will allow selecting more sentences before hitting the maximum word count, and will also make the Information Ordering module more effective. We would look into developing other experts such as Topical Closeness and precedence experts on top of the existing Chronological expert.

Despite LexRank sentence choices being qualitatively salient, much more so than first-sentences,

they did not always display good recall of events/concepts. Of X concepts dispersed amongst the original ten documents (ten per topic), sometimes LexRank will select only a small subset of X. For example, in a summarization of a murder, all the details of the actual murder are missing (see Section 6). This is a source of lack of coherence, as there are missing causal links in the chain, and also renders summarizations useless in practice. Summarization as a lossy compression system for transmitting large amounts of information in bitesize pieces should have strong objectives for concept and/or event recall. One of the methods of redundancy, event extraction, could also be used to penalize poor event recall. This will be a necessary metric to incorporate into future efforts, that is also algorithm independent.

5 Conclusion

In this work, we hypothesized that while the first-sentence summary would be a competitive baseline due to the inverse pyramid structure of the data, LexRank would outperform the naive heuristic. This was found to not be the case for our data as the first-sentence baseline outperformed all other baselines for nearly all ROUGE metrics.

We further hypothesized that using complex sentence representations via neural and distributional semantic embeddings trained on large corpora would improve LexRank sentence saliency, and this was also shown to not be the case, with TF-IDF outperforming all other variants.

Extensive experimentation with the system indicates that summary quality could be improved upon by improving discourse coherence, anaphora resolution and by penalizing redundancies to minimize paraphrases of the same salient topics. Additionally, modeling topic relevance in LexRank scores and removing bias towards long sentences in output could generate more content rich and representative summaries.

References

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. 2017. *Text summarization techniques: A brief survey.* arXiv preprint arXiv:1707.02268.

Baroni, M., Dinu, G., and Kruszewski, G 2014. *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors.* In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 238-247).

Bollegala, D., Okazaki, N., and Ishizuka, M. 2012. *A Preference Learning Approach to Sentence Ordering for Multi-document Summarization.* Information Sciences 217, 22, 78-95.

Brin, S., Page, L., Motwani, R., and Winograd, T. 1998. *The PageRank citation ranking: Bringing order to the web.* Stanford InfoLab.

Erkan, G., and Radev, D. R. 2004. *Lexrank: Graph-based lexical centrality as salience in text summarization.* Journal of Artificial Intelligence Research, 22, 457-479.

Kiss, T., and Strunk, J. 2006 *Unsupervised multilingual sentence boundary detection.* Computational Linguistics, 32(4), 485-525

Le, Q., and Mikolov, T. 2014 *Distributed representations of sentences and documents.* In International Conference on Machine Learning (pp. 1188-1196).

Levy, O., Goldberg, Y., and Dagan, I. 2015 *Improving distributional similarity with lessons learned from word embeddings.* Transactions of the Association for Computational Linguistics, 3, 211-225.

Nenkova, A., and McKeown, K. 2012. *A survey of text summarization techniques.* In Mining text data (pp. 43-76). Springer, Boston, MA.

Pennington, J., Socher, R., and Manning, C. 2014. *Glove: Global vectors for word representation* In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Porter, M. F. 1980 *An algorithm for suffix stripping.* Program, 14(3), 130-137.

6 Appendix

6.1 TF-IDF LexRank Examples

Note that all of the events of the actual murder are missing.

critics of prosecutors here said that they had little experience investigating and trying homicides in boulder, a college city that has about one murder a year.

at the same time, beckner said he was "excited "about new evidence.

so when the ramseys had three days of interviews last week with investigators from the district attorney's office, gordon wondered, "why the wait?

the ramsey interviews arrive as boulder, colo., district attorney alex hunter decides whether to take

the case to a grand jury.
“they may be posturing; they may be trying to help, ”mueller said.

6.2 Doc2Vec without stopword removal

When they came in February, shortly after their son’s killing, they attained near-celebrity status _ trailed by television cameras, greeted by crowds and transported from place to place by motorcade.
What happens next could intensify, rather than quell, the racial tensions that have been simmering in the two months since the shooting.

6.3 SpaCy without stopword removal

That case is set to begin on Tuesday.
and another falling down some steps as if he had been shot.
Friday, acting Justice Patricia Anne Williams, who refused defense requests to put off scheduling a date for the case, chose the Jan. 3 date after the defense lawyers said they would need time to go through the prosecution’s evidence and discuss possible pretrial motions.
It was a point that Worth disputed outside court by saying, “I don’t think I have a conflict of interest, though it is obvious that Kornberg would like for me to go.”

6.4 Doc2vec with stopword removal

The lawyers also said there could be pretrial scheduling conflicts because of other trials some of them are involved in, like the Abner Louima case, in which officers are accused of brutalizing a Haitian immigrant.
Police officials declined to say where the officers _ Kenneth Boss, Richard Murphy, Carroll and McMellon _ would be working, adding only that they were on modified duty and carrying no guns or badges.
They left the courtroom to the cheers of nearly 200 other officers, gathered outside the Bronx County Building.

6.5 SpaCy with stopword removal

“We are waiting for this, for justice,” he said, “and we feel strong.”
The four officers fired 41 shots, hitting Diallo 19 times.
The Rev. Al Sharpton, who organized the daily protests over the Diallo shooting and is an adviser to the family, also is expected to attend the meeting, said Steven Reed, Johnson’s spokesman.