Company-Oriented Extractive Summarization of Financial News[*]

Katja Filippova[†], Mihai Surdeanu[‡], Massimiliano Ciaramita[‡], Hugo Zaragoza[‡]

[†]EML Research gGmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany

[‡]Yahoo! Research
Avinguda Diagonal 177
08018 Barcelona, Spain

`filippova@eml-research.de,{mihais,massi,hugoz}@yahoo-inc.com`

Abstract

The paper presents a multi-document summarization system which builds company-specific summaries from a collection of financial news such that the extracted sentences contain novel and relevant information about the corresponding organization. The user's familiarity with the company's profile is assumed. The goal of such summaries is to provide information useful for the short-term trading of the corresponding company, i.e., to facilitate the inference from news to stock price movement in the next day. We introduce a novel query (i.e., company name) expansion method and a simple unsupervised algorithm for sentence ranking. The system shows promising results in comparison with a competitive baseline.

1 Introduction

Automatic text summarization has been a field of active research in recent years. While most methods are extractive, the implementation details differ considerably depending on *the goals* of a summarization system. Indeed, the intended use of the summaries may help significantly to adapt a particular summarization approach to a specific task whereas the broadly defined goal of preserving relevant, although generic, information may turn out to be of little use.

In this paper we present a system whose goal is to extract sentences from a collection of financial news to inform about important events concerning companies, e.g., to support trading (i.e., buy or sell) the corresponding symbol on the next day, or managing a portfolio. For example, a company's announcement of surpassing its earnings' estimate is likely to have a positive short-term effect on its stock price, whereas an announcement of job cuts is likely to have the reverse effect. We demonstrate how existing methods can be extended to achieve precisely this goal.

In a way, the described task can be classified as query-oriented multi-document summarization because we are mainly interested in information *related* to the company and its sector. However, there are also important differences between the two tasks.

- The name of the company is not a query, e.g., as it is specified in the context of the DUC competitions[1], and requires an extension. Initially, a query consists exclusively of the "*symbol*", i.e., the abbreviation of the name of a company as it is listed on the stock market. For example, *WPO* is the abbreviation used on the stock market to refer to *The Washington Post*–a large media and education company. Such symbols are rarely encountered in the news and cannot be used to find all the related information.

- The summary has to provide *novel* information related to the company and should avoid general facts about it which the user is supposed to know. This point makes the task related to update summarization where one has to provide the user with new information

[1]`http://duc.nist.gov`; since 2008 TAC: `http://www.nist.gov/tac`.

given some background knowledge[2]. In our case, general facts about the company are assumed to be known by the user. Given *WPO*, we want to distinguish between *The Washington Post is owned by The Washington Post Company, a diversified education and media company* and *The Post recently went through its third round of job cuts and reported an 11% decline in print advertising revenues for its first quarter*, the former being an example of background information whereas the latter is what we would like to appear in the summary. Thus, the similarity to the query alone is not the decisive parameter in computing sentence relevance.

- While the summaries must be specific for a given organization, important but general financial events that drive the overall market must be included in the summary. For example, the recent subprime mortgage crisis affected the entire economy regardless of the sector.

Our system proceeds in the three steps illustrated in Figure 1. First, the company symbol is expanded with terms relevant for the company, either directly – e.g., *iPod* is directly related to Apple Inc. – or indirectly – i.e., using information about the industry or sector the company operates in. We detail our symbol expansion algorithm in Section 3. Second, this information is used to rank sentences based on their relatedness to the expanded query and their overall importance (Section 4). Finally, the most relevant sentences are re-ranked based on the degree of novelty they carry (Section 5).

The paper makes the following contributions. First, we present a new query expansion technique which is useful in the context of company-dependent news summarization as it helps identify sentences important to the company. Second, we introduce a simple and efficient method for sentence ranking which foregrounds novel information of interest. Our system performs well in terms of the ROUGE score (Lin & Hovy, 2003) compared with a competitive baseline (Section 6).

2 Data

The data we work with is a collection of financial news consolidated and distributed by Yahoo! Finance[3] from various sources[4]. Each story is labeled as being relevant for a company – i.e., it appears in the company's RSS feed – if the story mentions either the company itself or the sector the company belongs to. Altogether the corpus contains 88,974 news articles from a period of about 5 months (148 days). Some articles are labeled as being relevant for several companies. The total number of *(company name, news collection)* pairs is 46,444.

The corpus is cleaned of HTML tags, embedded graphics and unrelated information (e.g., ads, frames) with a set of manually devised rules. The filtering is not perfect but removes most of the noise. Each article is passed through a language processing pipeline (described in (Atserias et al., 2008)). Sentence boundaries are identified by means of simple heuristics. The text is tokenized according to Penn TreeBank style and each token lemmatized using Wordnet's morphological functions. Part of speech tags and named entities (*LOC, PER, ORG, MISC*) are identified by means of a publicly available named-entity tagger[5] (Ciaramita & Altun, 2006, SuperSense). Apart from that, all sentences which are shorter than 5 tokens and contain neither nouns nor verbs are sorted out. We apply the latter filter as we are interested in textual information only. Numeric information contained, e.g., in tables can be easily and more reliably obtained from the indices tables available online.

3 Query Expansion

In company-oriented summarization query expansion is crucial because, by default, our query contains only the *symbol*, that is the abbreviation of the name of the company. Unfortunately, existing query expansion techniques which utilize such knowledge sources as WordNet or Wikipedia are not useful for symbol expansion. WordNet does not include organizations in any systematic way. Wikipedia covers many companies but it is unclear how it can be used for expansion.

---

[2]See the DUC 2007 and 2008 update tracks.

[3]http://finance.yahoo.com

[4]http://biz.yahoo.com, http://www.seekingalpha.com, http://www.marketwatch.com, http://www.reuters.com, http://www.fool.com, http://www.thestreet.com, http://online.wsj.com, http://www.forbes.com, http://www.cnbc.com, http://us.ft.com, http://www.minyanville.com

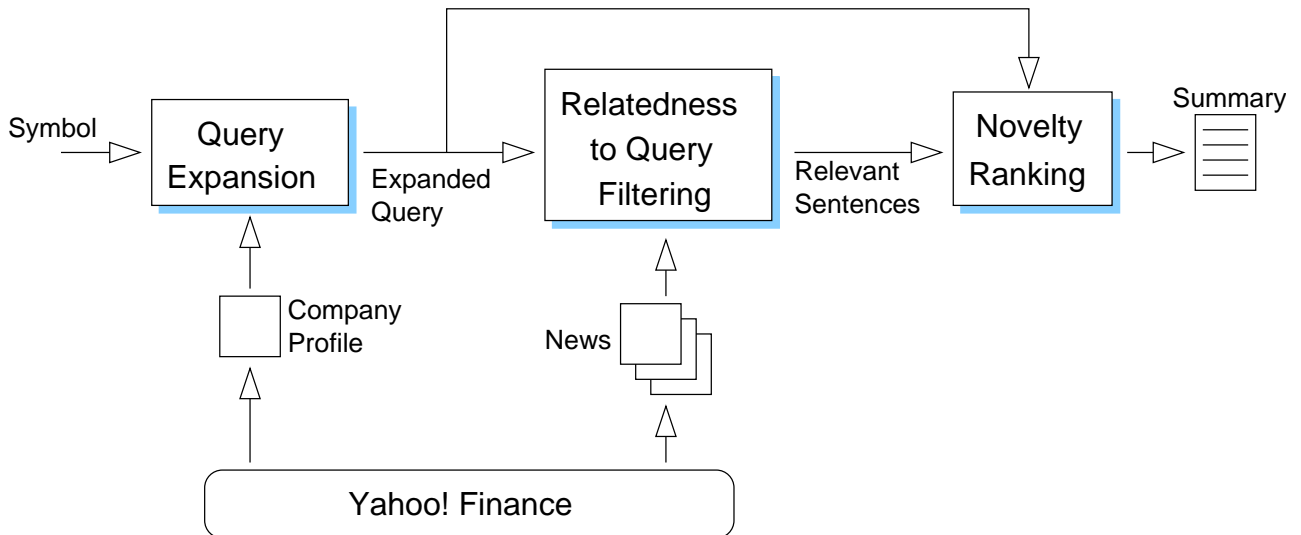[5]http://sourceforge.net/projects/supersensetag

Figure 1: System architecture

Intuitively, a good expansion method should provide us with a list of products, or properties, of the company, the field it operates in, the typical customers, etc. Such information is normally found on the profile page of a company at Yahoo! Finance[6]. There, so called "business summaries" provide succinct and financially relevant information about the company. Thus, we use business summaries as follows. For every company symbol in our collection, we download its business summary, split it into tokens, remove all words but nouns and verbs which we then lemmatize. Since words like *company* are fairly uninformative in the context of our task, we do not want to include them in the expanded query. To filter out such words, we compute the company-dependent TF*IDF score for every word on the collection of all business summaries:

$$\text{score}(w) = \text{tf}_{w,c} \times \log\left(\frac{N}{\text{cf}_w}\right) \qquad (1)$$

where c is the business summary of a company, $\text{tf}_{w,c}$ is the frequency of w in c, N is the total number of business summaries we have, cf_w is the number of summaries that contain w. This formula penalizes words occurring in most summaries (e.g., *company, produce, offer, operate, found, headquarter, management*). At the moment of running the experiments, N was about 3,000, slightly less than the total number of sym-

bols because some companies do not have a business summary on Yahoo! Finance. It is important to point out that companies without a business summary are usually small and are seldom mentioned in news articles: for example, these companies had relevant news articles in only 5% of the days monitored in this work.

Table 1 gives the ten high scoring words for three companies (Apple Inc. – the computer and software manufacture, Delta Air Lines – the airline, and DaVita – dyalisis services). Table 1 shows that this approach succeeds in expanding the symbol with terms directly related to the company, e.g., *ipod* for Apple, but also with more general information like the industry or the company operates in, e.g., *software* and *computer* for Apple. All words whose TF*IDF score is above a certain threshold θ are included in the expanded query (θ was tuned to a value of 5.0 on the development set).

4 Relatedness to Query

Once the expanded query is generated, it can be used for sentence ranking. We chose the system of Otterbacher et al. (2005) as a a starting point for our approach and also as a competitive baseline because it has been successfully tested in a similar setting–it has been applied to multi-document query-focused summarization of news documents.

Given a graph $G = (S, E)$, where S is the set of all sentences from all input documents, and E is the set of edges representing normalized sentence similarities, Otterbacher et al. (2005) rank all sen-

---

[6]http://finance.yahoo.com/q/pr?s=AAPL where the trading symbol of any company can be used instead of AAPL.

AAPL	DAL	DVA
apple	air	dialysis
music	flight	davita
mac	delta	esrd
software	lines	kidney
ipod	schedule	inpatient
computer	destination	outpatient
peripheral	passenger	patient
movie	cargo	hospital
player	atlanta	disease
desktop	fleet	service

Table 1: Top 10 scoring words for three companies

tence nodes based on the inter-sentence relations as well as the relevance to the query q. Sentence ranks are found iteratively over the set of graph nodes with the following formula:

$$r(s, q) = \lambda \frac{\text{rel}(s|q)}{\sum_{t \in S} \text{rel}(t|q)} + (1-\lambda) \sum_{t \in S} \frac{\text{sim}(s, t)}{\sum_{v \in S} \text{sim}(v, t)} r(t, q) \quad (2)$$

The first term represents the importance of a sentence defined in respect to the query, whereas the second term infers the importance of the sentence from its relation to other sentences in the collection. $\lambda \in (0, 1)$ determines the relative importance of the two terms and is found empirically. Another parameter whose value is determined experimentally is the sentence similarity threshold τ, which determines the inclusion of a sentence in G. Otterbacher et al. (2005) report 0.2 and 0.95 to be the optimal values for τ and λ respectively. These values turned out to produce the best results also on our development set and were used in all our experiments. Similarity between sentences is defined as the cosine of their vector representations:

$$\text{sim}(s, t) = \frac{\sum_{w \in s \cap t} \text{weight}(w)^2}{\sqrt{\sum_{w \in s} \text{weight}(w)^2} \times \sqrt{\sum_{w \in t} \text{weight}(w)^2}} \quad (3)$$

$$\text{weight}(w) = \text{tf}_{w,s} \text{idf}_{w,S} \quad (4)$$

$$\text{idf}_{w,S} = \log\left(\frac{|S| + 1}{0.5 + \text{sf}_w}\right) \quad (5)$$

where $\text{tf}_{w,s}$ is the frequency of w in sentence s, $|S|$ is the total number of sentences in the documents from which sentences are to be extracted, and sf_w is the number of sentences which contain the word w (all words in the documents as well

as in the query are stemmed and stopwords are removed from them). Relevance to the query is defined in Equation (6) which has been previously used for sentence retrieval (Allan et al., 2003):

$$\text{rel}(s|q) = \sum_{w \in q} \log(\text{tf}_{w,s} + 1) \times \log(\text{tf}_{w,q} + 1) \times \text{idf}_{w,S} \quad (6)$$

where $\text{tf}_{w,x}$ stands for the number of times w appears in x, be it a sentence (s) or the query (q). If a sentence shares no words other than stopwords with the query, the relevance becomes zero. Note that without the relevance to the query part Equation 2 takes only inter-sentence similarity into account and computes the weighted PageRank (Brin & Page, 1998).

In defining the relevance to the query, in Equation (6), words which do not appear in too many sentences in the document collection weigh more. Indeed, if a word from the query is contained in many sentences, it should not count much. But it is also true that not all words from the query are equally important. As it has been mentioned in Section 3, words like *product* or *offer* appear in many business summaries and are equally related to any company. To penalize such words, when computing the relevance to the query, we multiply the relevance score of a given word w with the inverted document frequency of w on the corpus of business summaries $Q - \text{idf}_{w,Q}$:

$$\text{idf}_{w,Q} = \log\left(\frac{|Q|}{\text{qf}_w}\right) \quad (7)$$

We also replace $\text{tf}_{w,s}$ with the indicator function $s(w)$ since it has been reported to be more adequate for sentences, in particular for sentence alignment (Nelken & Shieber, 2006):

$$s(w) = \begin{cases} 1 & \text{if } s \text{ contains } w \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Thus, the modified formula we use to compute sentence ranks is as follows:

$$\text{rel}(s|q) = \sum_{w \in q} s(w) \times \log(\text{tf}_{w,q} + 1) \times \text{idf}_{w,S} \times \text{idf}_{w,Q} \quad (9)$$

We call these two ranking algorithms that use the formula in (2) OTTERBACHER and QUERY WEIGHTS, the difference being the way the relevance to the query is computed: (6) or (9). We use the OTTERBACHER algorithm as a baseline in the experiments reported in Section 6.

5 Novelty Bias

Apart from being related to the query, a good summary should provide the user with novel information. According to Equation (2), if there are, say, two sentences which are highly similar to the query and which share some words, they are likely to get a very high score. Experimenting with the development set, we observed that sentences about the company, such as e.g., *DaVita, Inc. is a leading provider of kidney care in the United States, providing dialysis services and education for patients with chronic kidney failure and end stage renal disease*, are ranked high although they do not contribute new information. However, a non-zero similarity to the query is indeed a good filter of the information related to the company and to its sector and can be used as a prerequisite of a sentence to be included in the summary. These observations motivate our proposal for a ranking method which aims at providing relevant *and* novel information at the same time.

Here, we explore two alternative approaches to add the novelty bias to the system:

- The first approach bypasses the relatedness to query step introduced in Section 4 completely. Instead, this method merges the discovery of query relatedness and novelty into a single algorithm, which uses a sentence graph that contains edges *only* between sentences related to the query, (i.e., sentences for which $rel(s|q) > 0$). All edges connecting sentences which are unrelated to the query are skipped in this graph. In this way we limit the novelty ranking process to a subset of sentences related to the query.

- The second approach models the problem in a re-ranking architecture: we take the top ranked sentences after the relatedness-to-query filtering component (Section 4) and re-rank them using the novelty formula introduced below.

The main difference between the two approaches is that the former uses relatedness-to-query and novelty information but ignores the overall importance of a sentence as given by the PageRank algorithm in Section 4, while the latter combines all these aspects –i.e., importance of sentences, relatedness to query, and novelty– using the re-ranking architecture.

To amend the problem of general information ranked inappropriately high, we modify the word-weighting formula (4) so that it implements a novelty bias, thus becoming dependent on the query. A straightforward way to define the novelty weight of a word would be to draw a line between the "known" words, i.e., words appearing in the business summary, and the rest. In this approach all the words from the business summary are equally related to the company and get the weight of 0:

$$\text{weight}(w) = \begin{cases} 0 & \text{if } Q \text{ contains } w \\ \text{tf}_{w,s}\text{idf}_{w,S} & \text{otherwise} \end{cases} \tag{10}$$

We call this weighting scheme SIMPLE. As an alternative, we also introduce a more elaborate weighting procedure which incorporates the relatedness-to-query (or rather distance from query) in the word weight formula. Intuitively, the more related to the query a word is (e.g., *DaVita*, the name of the company), the more familiar to the user it is and the smaller its novelty contribution is. If a word does not appear in the query at all, its weight becomes equal to the usual $\text{tf}_{w,s}\text{idf}_{w,S}$:

$$\text{weight}(w) = \left(1 - \frac{\text{tf}_{w,q} \times \text{idf}_{w,Q}}{\sum_{w_i \in q} \text{tf}_{w_i,q} \times \text{idf}_{w_i,Q}}\right) \times \text{tf}_{w,s}\text{idf}_{w,S} \tag{11}$$

The overall novelty ranking formula is based on the query-dependent PageRank introduced in Equation (2). However, since we already incorporate the relatedness to the query in these two settings, we focus only on related sentences and thus may drop the relatedness to the query part from (2):

$$\text{r}'(s,q) = \lambda + (1-\lambda) \sum_{t \in S} \frac{\text{sim}(s,t,q)}{\sum_{u \in S} \text{sim}(t,u,q)} \tag{12}$$

We set λ to the same value as in OTTERBACHER. We deliberately set the sentence similarity threshold τ to a very low value (0.05) to prevent the graph from becoming exceedingly bushy. Note that this novelty-ranking formula can be equally applied in both scenarios introduced at the beginning of this section. In the first scenario, S stands for the set of nodes in the graph that contains only sentences related to the query. In the second scenario, S contains the highest ranking sentences detected by the relatedness-to-query component (Section 4).

5.1 Redundancy Filter

Some sentences are repeated several times in the collection. Such repetitions, which should be avoided in the summary, can be filtered out either before or after the sentence ranking. We apply a simple repetition check when incrementally adding ranked sentences to the summary. If a sentence to be added is almost identical to the one already included in the summary, we skip it. Identity check is done by counting the percentage of non-stop word lemmas in common between two sentences. 95% is taken as the threshold.

We do not filter repetitions *before* the ranking has taken place because often such repetitions carry important and relevant information. The redundancy filter is applied to all the systems described as they are equally prone to include repetitions.

6 Evaluation

We randomly selected 23 company stock names, and constructed a document collection for each containing all the news provided in the Yahoo! Finance news feed for that company in a period of two days (the time period was chosen randomly). The average length of a news collection is about 600 tokens. When selecting the company names, we took care of not picking those which have only a few news articles for that period of time. This resulted into 9.4 news articles per collection on average. From each of these, three human annotators independently selected up to ten sentences. All annotators had average to good understanding of the financial domain. The annotators were asked to choose the sentences which could best help them decide whether to buy, sell or retain stock for the company the following day and present them in the order of decreasing importance. The annotators compared their summaries of the first four collections and clarified the procedure before proceeding with the other ones. These four collections were then later used as a development set.

All summaries – manually as well as automatically generated – were cut to the first 250 words which made the summaries 10 words shorter on average. We evaluated the performance automatically in terms of ROUGE-2 (Lin & Hovy, 2003) using the parameters and following the methodology from the DUC events. The results are presented in Table 2. We also report the 95% confidence intervals in brackets. As in DUC, we used

METHOD	ROUGE-2
Otterbacher	0.255 (0.226 - 0.285)
Query Weights	0.289 (0.254 - 0.324)
Novelty Bias (simple)	**0.315** (0.287 - 0.342)
Novelty Bias	0.302 (0.277 - 0.329)
Manual	0.472 (0.415 - 0.531)

Table 2: Results of the four extraction methods and human annotators

jackknife for each *(query, summary)* pair and computed a macro-average to make human and automatic results comparable (Dang, 2005). The scores computed on summaries produced by humans are given in the bottom line (MANUAL) and serve as upper bound and also as an indicator for the inter-annotator agreement.

6.1 Discussion

From Table 2 follows that the modifications we applied to the baseline are sensible and indeed bring an improvement. QUERY WEIGHTS performs better than OTTERBACHER and is in turn outperformed by the algorithms biased to novel information (the two NOVELTY systems). The overlap between the confidence intervals of the baseline and the simple version of the novelty algorithm is minimal (0.002).

It is remarkable that the achieved improvement is due to a more balanced relatedness to the query ranking (9), as well as to the novelty bias re-ranking. The fact that the simpler novelty weighting formula (10) produced better results than the more elaborated one (11) requires a deeper analysis and a larger test set to explain the difference. Our conjecture so far is that the SIMPLE approach allows for a better combination of both novelty and relatedness to query. Since the more complex novelty ranking formula penalizes terms related to the query (Equation (11)), it favors a scenario where novelty is boosted in detriment of relatedness to query, which is not always realistic.

It is important to note that, compared with the baseline, we did not do any parameter tuning for λ and the inter-sentence similarity threshold. The improvement between the system of Otterbacher et al. (2005) and our best model is statistically significant.

6.2 System Combination

Recall from Section 5 that the motivation for promoting novel information came from the fact that sentences with background information about the company obtained very high scores: they were related but not novel. The sentences ranked by OTTERBACHER or QUERY WEIGHTS required a re-ranking to include related *and* novel sentences in the summary. We checked whether novelty re-ranking brings an improvement if added on top of a system which does not have a novelty bias (baseline or QUERY WEIGHTS) and compared it with the setting where we simply limit the novelty ranking to all the sentences related to the query (NOVELTY SIMPLE and NOVELTY). In the similarity graph, we left only edges between the first 30 sentences from the ranked list produced by one of the two algorithms described in Section 4 (OTTERBACHER or QUERY WEIGHTS). Then we ranked the sentences biased to novel information the same way as described in Section 5. The results are presented in Table 3. What we evaluate here is whether a combination of two methods performs better than the simple heuristics of discarding edges between sentences unrelated to the query.

METHOD	ROUGE-2
Otterbacher + Novelty simple	**0.280** (0.254 - 0.306)
Otterbacher + Novelty	**0.273** (0.245 - 0.301)
Query Weights + Novelty simple	0.275 (0.247 - 0.302)
Query Weights + Novelty	0.265 (0.242 - 0.289)

Table 3: Results of the combinations of the four methods

From the four possible combinations, there is an improvement over the baseline only (0.255 vs. 0.280 resp. 0.273). None of the combinations performs better than the simple novelty bias algorithm on a subset of edges. This experiment suggests that, at least in the scenario investigated here (short-term monitoring of publicly-traded companies), novelty is more important than relatedness to query. Hence, the simple novelty bias algorithm, which emphasizes novelty and incorporates relatedness to query only through a loose constraint ($rel(s|q) > 0$) performs better than complex models, which are more constrained by the relatedness to query.

7 Related Work

Summarization has been extensively investigated in recent years and to date there exists a multitude of very different systems. Here, we review those that come closest to ours in respect to the task and that concern extractive multi-document query-oriented summarization. We also mention some work on using textual news data for stock indices prediction which we are aware of.

Stock market prediction: Wüthrich et al. (1998) were among the first who introduced an automatic stock indices prediction system which relies on textual information only. The system generates weighted rules each of which returns the probability of the stock going up, down or remaining steady. The only information used in the rules is the presence or absence of certain keyphrases provided by a human expert who *"judged them to be influential factors potentially moving stock markets"*. In this approach, training data is required to measure the usefulness of the keyphrases for each of the three classes. More recently, Lerman et al. (2008) introduced a forecasting system for prediction markets that combines news analysis with a price trend analysis model. This approach was shown to be successful for the forecasting of public opinion about political candidates in such prediction markets. Our approach can be seen as a complement to both these approaches, necessary especially for financial markets where the news typically cover many events, only some related to the company of interest.

Unsupervised summarization systems extract sentences whose relevance can be inferred from the inter-sentence relations in the document collection. In (Radev et al., 2000), the centroid of the collection, i.e., the words with the highest TF*IDF, is considered and the sentences which contain more words from the centroid are extracted. Mihalcea & Tarau (2004) explore several methods developed for ranking documents in information retrieval for the single-document summarization task. Similarly, Erkan & Radev (2004) apply in-degree and PageRank to build a summary from a collection of related documents. They show that their method, called LexRank, achieves good results. In (Otterbacher et al., 2005; Erkan, 2006) the ranking function of LexRank is extended to become applicable to query-focused summarization. The rank of a sentence is determined not just by its relation to other sentences in

the document collection but also by its relevance to the query. Relevance to the query is defined as the word-based similarity between query and sentence.

Query expansion has been used for improving information retrieval (IR) or question answering (QA) systems with mixed results. One of the problems is that the queries are expanded word by word, ignoring the context and as a result the extensions often become inadequate[7]. However, Riezler et al. (2007) take the entire query into account when adding new words by utilizing techniques used in statistical machine translation.

Query expansion for summarization has not yet been explored as extensively as in IR or QA. Nastase (2008) uses Wikipedia and WordNet for query expansion and proposes that a concept can be expanded by adding the text of all hyperlinks from the first paragraph of the Wikipedia article about this concept. The automatic evaluation demonstrates that extracting relevant concepts from Wikipedia leads to better performance compared with WordNet: both expansion systems outperform the no-expansion version in terms of the ROUGE score. Although this method proved helpful on the DUC data, it seems less appropriate for expanding company names. For small companies there are short articles with only a few links; the first paragraphs of the articles about larger companies often include interesting rather than relevant information. For example, the text preceding the contents box in the article about Apple Inc. (AAPL) states that *"Fortune magazine named Apple the most admired company in the United States"*[8]. The link to the article about the Fortune magazine can be hardly considered relevant for the expansion of AAPL. Wikipedia category information, which has been successfully used in some NLP tasks (Ponzetto & Strube, 2006, inter alia), is too general and does not help discriminate between two companies from the same sector.

Our work suggests that query expansion is needed for summarization in the financial domain. In addition to previous work, we also show that another key factor for success in this task is detecting and modeling the novelty of the target content.

---

[7]E.g., see the proceedings of TREC 9, TREC 10: `http://trec.nist.gov`.

[8]Checked on September 17, 2008.

8 Conclusions

In this paper we presented a multi-document company-oriented summarization algorithm which extracts sentences that are both relevant for the given organization and novel to the user. The system is expected to be useful in the context of stock market monitoring and forecasting, that is, to help the trader predict the move of the stock price for the given company. We presented a novel query expansion method which works particularly well in the context of company-oriented summarization. Our sentence ranking method is unsupervized and requires little parameter tuning. An automatic evaluation against a competitive baseline showed supportive results, indicating that the ranking algorithm is able to select relevant sentences and promote novel information at the same time.

In the future, we plan to experiment with positional features which have proven useful for generic summarization. We also plan to test the system extrinsically. For example, it would be of interest to see if a classifier may predict the move of stock prices based on a set of features extracted from company-oriented summaries.

References

Allan, James, Courtney Wade & Alvaro Bolivar (2003). Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* Toronto, On., Canada, 28 July – 1 August 2003, pp. 314–321.

Atserias, Jordi, Hugo Zaragoza, Massimiliano Ciaramita & Giuseppe Attardi (2008). Semantically annotated snapshot of the English Wikipedia. In *Proceedings of the 6th International Conference on Language Resources and Evaluation,* Marrakech, Morocco, 26 May – 1 June 2008.

Brin, Sergey & Lawrence Page (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Ciaramita, Massimiliano & Yasemin Altun (2006). Broad-coverage sense disambiguation

and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,* Sydney, Australia, 22–23 July 2006, pp. 594–602.

Dang, Hoa Trang (2005). Overview of DUC 2005. In *Proceedings of the 2005 Document Understanding Conference held at the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 9–10 October 2005.

Erkan, Güneş (2006). Using biased random walks for focused summarization. In *Proceedings of the 2006 Document Understanding Conference held at the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,,* New York, N.Y., 8–9 June 2006.

Erkan, Güneş & Dragomir R. Radev (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Lerman, Kevin, Ari Gilder, Mark Dredze & Fernando Pereira (2008). Reading the markets: Forecasting public opinion of political candidates by news analysis. In *Proceedings of the 22st International Conference on Computational Linguistics,* Manchester, UK, 18–22 August 2008, pp. 473–480.

Lin, Chin-Yew & Eduard H. Hovy (2003). Automatic evaluation of summaries using N-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Edmonton, Alberta, Canada, 27 May –1 June 2003, pp. 150–157.

Mihalcea, Rada & Paul Tarau (2004). Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing,* Barcelona, Spain, 25–26 July 2004, pp. 404–411.

Nastase, Vivi (2008). Topic-driven multi-document summarization with encyclopedic knowledge and activation spreading. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing,* Honolulu, Hawaii, 25–27 October 2008. To appear.

Nelken, Rani & Stuart M. Shieber (2006). Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics,* Trento, Italy, 3–7 April 2006, pp. 161–168.

Otterbacher, Jahna, Güneş Erkan & Dragomir Radev (2005). Using random walks for question-focused sentence retrieval. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pp. 915–922.

Ponzetto, Simone Paolo & Michael Strube (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* New York, N.Y., 4–9 June 2006, pp. 192–199.

Radev, Dragomir R., Hongyan Jing & Malgorzata Budzikowska (2000). Centroid-based summarization of mutliple documents: Sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the Workshop on Automatic Summarization at ANLP/NAACL 2000,* Seattle, Wash., 30 April 2000, pp. 21–30.

Riezler, Stefan, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal & Yi Liu (2007). Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* Prague, Czech Republic, 23–30 June 2007, pp. 464–471.

Wüthrich, B, D. Permunetilleke, S. Leung, V. Cho, J. Zhang & W. Lam (1998). Daily prediction of major stock indices from textual WWW data. In *In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining - KDD-98*, pp. 364–368.