NUS at TAC 2008: Augmenting Timestamped Graphs with Event Information and Selectively Expanding Opinion Contexts

Ziheng Lin, Huu Hung Hoang,
Long Qiu, Shiren Ye and **Min-Yen Kan***
School of Computing
National University of Singapore
Singapore, 117590
{linzihen|hoanghuu|qiul|yesr|kanmy}@comp.nus.edu.sg

Abstract

Our team explored several new approaches in the update and opinion summarization tasks in the Text Analysis Conference (TAC) 2008. For the update task, we refine our previous timestamped graph approach by incorporating information about temporal ordering of events in the articles, by using the publicly available Tarsqi toolkit. For the pilot opinion task, we utilize the provided opinion snippets as clues to locate their source sentences. Our system expands the context around the snippets' source sentences for two purposes. First, to more accurately identify the polarity of the contained opinion; second, to selectively include the context for added coherence in our answers.

1 Introduction

This year, the Text Analysis Conference (TAC) 2008 summarization track featured two summarization tasks as testbeds. One is the *update summarization* task, piloted in DUC 2007 (DUC, 2007). Different from the traditional multi-document summarization, an update summary aims to highlight new information to the user who is already aware of certain background information. This background might include facts about the topic that have been covered in earlier documents. Additional challenges thus include differentiating such background information from new content, as well as deciding whether part of the previous news needs to be recapped.

DUC 2007 participants experimented with various approaches to generating update summaries. Individual terms were the focus of the analysis and sentences containing novel terms were generally favored, as represented by (Pingali et al., 2007). It was also noted that the importance of topical terms should not be downplayed naïvely simply because they were repeated in the update document set (Kolla et al., 2007). In other work, Summary Content Units (SCUs) gleaned over past DUC data were used to train machine learners for sentence selection (Copeck et al., 2007).

TAC 2008 pilots a new summarization task – *opinion summarization*, where systems are required to analyze blog articles and summarize the opinions in them. A hallmark of this task is the subtask of opinion recognition. Also, as the input is in the form of blog posts, it is potentially noisy and ungrammatical. As such, this special multi-document summarization task also poses unique problems.

In DUC 2007, we applied a timestamped graph (TSG) formalism to the update summarization task and had satisfactory preliminary results. In this year's work, we augmented our

*Funding for this research was generously provided by a Interactive and Digital Media Grant NRF2007IDM-IDM002-047 from the National Research Foundation of Singapore.

update summarization system by incorporating the Tarsqi toolkit (Saurí et al., 2005; Mani and Wilson, 2000; Mani et al., 2003), which tags and temporally orders (mentions of) events in texts. To our knowledge, our work reports the first time that temporal relations among events are considered in update summarization. Although our system did not perform well in the generic summarization task for document set A, our approach focused on the update summarization and resulted in a large improvement when summarizing for updates on the document set B (an improvement of 20 positions in ROUGE-2). This implies that temporal relations among events are helpful in extracting update information.

For the opinion summarization task, we argue that the *contexts* of mentions of opinions are important as they complement the points conveyed through the opinion mentions themselves. Accordingly, we focus on discerning and selectively incorporating the pertinent contexts of the given relevant text snippets. To this end, we mine terms reflecting users' interests from the Web and compile patterns commonly used to express supporting information. The experimental results validate our focus on the contexts: while a strategy to include simply the surrounding sentences makes a strong baseline opinion summarizer, carefully filtering such candidate contexts can improve the performance even further.

The remainder of this paper is organized as follows. Section 2 gives an overview of our update and opinion summarization systems. The technical details and evaluation results of them are presented in Section 3 and Section 4, respectively. Finally, Section 5 concludes the outcome of our participation this year.

2 System Overview

Update Summarization. A system overview of our update summarization system is shown in Figure 1. The *Temporal Link Extractor* based on Tarsqi labels events and time in the input and extracts temporal links among them. Given sentences labeled with event, time and temporal link information, the *Timestamp Tagger* assigns each sentence with a timestamp.

The next three modules are identical to our system reported in last year. The *Graph Constructor* adds sentences as nodes into our graph representation of the articles by the order of the sentence timestamps, and edges are constructed by computing sentence similarities. After the graph is constructed, the *Sentence Ranker* applies a PageRank algorithm to redistribute the sentence weights, resulting in a ranked list of the sentences after PageRank converges. With this ranked list, the *Sentence Extractor* uses a modified MMR reranker to extract highly-ranked sentences that are not overlapped with sentences from the summary for previous set and sentences that are just extracted.

Our system acts as a generic summarizer with Set A. When summarizing Set B, the previous summary for Set A is used as an input to the reranking process.

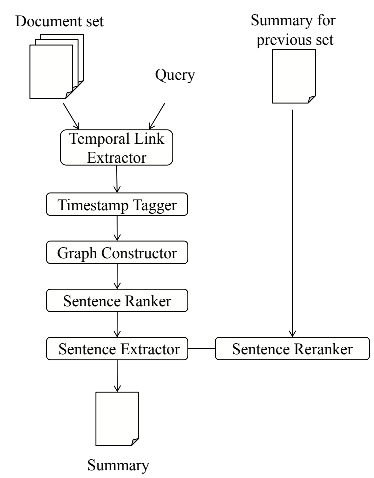

Figure 1: System overview for update summarizer

Opinion Summarization. Figure 2 gives an overview of our opinion summarization pipeline. The input to the system is a list of snippets, a set of blog documents, and a target that consists of multiple queries. For each given snippet, a *Snippet Context Selector* finds a sentence from the documents that has the maximum cosine similarity with the snippet, and selectively includes its previous and

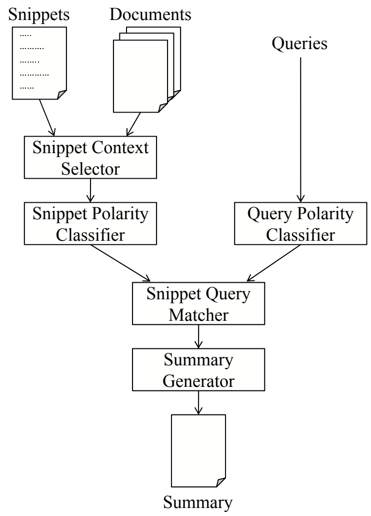

Figure 2: System overview for opinion summarizer

next sentences to produce an expanded snippet. The *Snippet Polarity Classifier* and *Query Polarity Classifier* examine the polarities of the expanded snippets and the queries, respectively. For a given target, each query will be matched with multiple expanded snippets by the *Snippet Query Matcher*. The matched sentences are then synthesized by the *Summary Generator* to produce a summary for each query.

The following two sections give an detailed description of our systems. In both systems, we treat the summarization task as extractive sentence selection (selecting k from N input sentences), and have specifically not attempted sentence post-editing.

3 An Augmented Timestamped Graph Model for Update Summarization

Standard graph-based approaches to text summarization, such as TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004) model the input set of documents as a graph, where nodes are document sentences and edges are drawn between sentences if they are somehow related. For example, if sentences are interpreted as bags of words, edges may be weighted due to undirected symmetric cosine similarity or directed, asymmetric Kullback-Leibler divergence. Subsequently, graph propagation may be applied to the graph (such as PageRank) and highly-weighted sentences are selected for inclusion in the output summary.

However, these methods assume a static graph model where all edges and nodes are present, which does not model how the input text emerges as nodes in the graph. A suitable evolutionary graph model that is related to human writing and reading processes may impart a better understanding of the text and improve the subsequent summarization process. Although such human processes vary widely, when we limit ourselves to expository texts, we find that both skilled writers and readers often follow conventional rhetorical styles (Endres-Niggemeyer, 1998; Liddy, 1991).

In our previous approach (Lin et al., 2007), we made two simple assumptions of human writing and reading processes: (1) writers write articles from the first sentence to the last, and (2) readers read articles from the first sentence to the last. These two naïve assumptions give a timestamp to each sentence, based on its sentence order in the text. For example, the first and sixth sentences of a text are assigned the respective timestamps of 1 and 6. The timestamps imply how the sentences can evolve in the graph. We add sentences into the graph in this chronological order – we add the first sentence, followed by the second sentence, and so forth, until the last sentence is added – and at each timestep consider which edges to add to the existing graph. In the case of multi-document input, we simply evolve the whole graph in parallel, as n single document instances. One simple method introduces all of the i^{th} sentences from all input documents into the graph in the i^{th} timestep. Figure 3 shows the graph building process that is used in (Lin et al., 2007). In our system, edge weights are measured by *concept similarity* (Ye et al., 2005) between two sentences.

Once a timestamped graph is built, we run a PageRank algorithm that is integrated with weighted edges and query-as-topic sensitivity on the graph to redistribute the node weights. When the ranking process converges, the node weights are output as sentence scores. A modified maximum marginal relevance (MMR)

Input: **M**, a cluster of m documents relating to a common event;
Variables:
i = index to sentences, initially 1;
G = the timestamped graph, initially empty.
Loop:

1. Add the i^{th} sentence of all documents into **G**. If a document is shorter than i, no sentence is added for this document.

2. Connect each existing sentence s in **G** to **one** other existing sentence in **G** that has the highest concept similarity with s and is not previously connected with s.

3. **if** there are no new sentences to add, break;
   else **i**++, go to Step 1.

Output: **G**, a timestamped graph.

Figure 3: Pseudocode for a timestamped graph construction algorithm

reranker is then applied to remove redundancy and extract summary sentences.

3.1 Tagging Timestamps with Event Ordering

In our previous work, the human writing and reading processes are approximated by the two naïve assumptions, which result in simply tagging the sentences with timestamps identical to their sentence order in the text. Our motivation was to evolve the text in a way similar to the human reading and writing processes. However, as we are dealing with news articles which often describe the reporting and updating of events, we believe that the actual news events' ordering is a more natural process to approximate the graph evolution process. The key focus of our work in this year's update task was to incorporate event based temporal information into our processes.

There have been many publications devoted to developing temporal awareness and reasoning systems. Sauri et al. (2005) developed a tool to automatically locate and tag all event-referring expressions in the input text. Mani and Wilson (2000) developed a tool for "recognizing the extents and normalized values of time expressions". Mani et al. (2003) developed a tagger that uses hand-crafted syntactic and lexical rules to label temporal links among time and event expressions. Tarsqi [1] is a toolkit that is composed of several temporal reasoning tools, including the abovementioned three. It is released as part of the AQUAINT program and fulfills our needs precisely. Tarsqi recognizes and tags event and time expressions using TimeML syntax in input texts, and attempts to infer the temporal ordering of the events within texts. Figures 4, 5 and 6 show a motivating example, in which the Tarsqi toolkit was applied to tag the news article XIN_ENG_20060123.0113 with events. The inferred temporal ordering of these events is shown in Figure 5. A forward link from events ei to ej means that ei happened before ej in timeline. As our text units are sentences and not events, we ignore the intra-sentential links and extract only the inter-sentential links, which result in the temporal ordering of the example sentences in Figure 6(a). The timestamps of the sentences can then be tagged with the sentence ordering.

s_1 A Turkish court has [dropped]$_{e1}$ [charges]$_{e2}$ against renowned novelist Orhan Pamuk for [insulting]$_{e3}$ the Turkish identity, local newspaper HURRIYET (Freedom) [reported]$_{e4}$ on Monday.

s_2 Pamuk's lawyer has [confirmed]$_{e5}$ the [dismissal]$_{e6}$ of the case.

s_3 The Istanbul court's [decision]$_{e7}$ [came]$_{e8}$ after the Justice Ministry [said]$_{e9}$ that the court [had]$_{e10}$ no authority to [try]$_{e11}$ him.

s_4 The court had [adjourned]$_{e12}$ Pamuk's [trial]$_{e13}$ shortly after it [began]$_{e14}$ on Dec. 16 and [asked]$_{e15}$ the Justice Ministry for a legal opinion on whether he could be [tried]$_{e16}$ under a new penal code.

s_5 Best-selling novelist Pamuk, 53, was [charged]$_{e17}$ with [insulting]$_{e18}$ the Turkish identity by [saying]$_{e19}$ that nobody in Turkey [dared]$_{e20}$ [mention]$_{e21}$ the [killing]$_{e22}$ of a million Armenians by the Ottoman Empire during World War I. The case has [catched]$_{e23}$ international limelight and the European Union has [voiced]$_{e24}$ grave concerns over it.

s_6 The EU [started]$_{e25}$ [talks]$_{e26}$ with Ankara on its entry bid in October.

s_7 The [negotiations]$_{e27}$ are [expected]$_{e28}$ to [last]$_{e29}$ at least a decade and the EU demands Turkey to [carry]$_{e30}$ out wide-ranging [reforms]$_{e31}$.

s_8 Turkey has [rejected]$_{e32}$ [allegations]$_{e33}$ of [genocide]$_{e34}$ against Armenians during World War I.

Figure 4: News report XIN_ENG_20060123.0113 in Set D0848H-B with events tagged.

Temporal Link Extractor: We first use Tarsqi to perform extraction of events and labeling of the temporal ordering of events, ignoring intra-sentential links and extracting only the temporal ordering of sentences using inter-

[1] http://www.timeml.org/site/tarsqi/toolkit/index.html

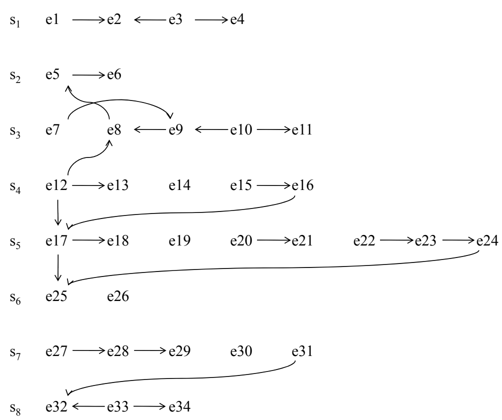

Figure 5: Temporal ordering of events of the news report in Figure 4.

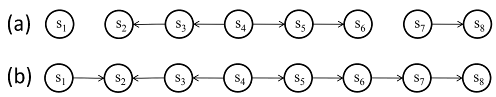

Figure 6: Temporal ordering of sentences extracted from Figure 5.

sentential links. As we are dealing with multiple documents, we first concatenate these documents together by the document timestamps shown in the file names (e.g., 20060123 in XIN_ENG_20060123.0113) before the system sends the concatenated document to Tarsqi. As shown in Figure 6(a), the resulting graph often is sparse – forward or backward links between any two consecutive sentences s_i and s_{i+1} may not be inferred if supporting evidence is not present. In these cases, we revert back to our default reading and writing model from our previous work and add a forward link from s_i to s_{i+1} (see Figure 6(b)). Note that a sentence s_i may have links pointing to non-consecutive sentences before and after it.

Timestamp Tagger: Using the sentence ordering extracted by the *Temporal Link Extractor*, we run a breadth-first search to tag these sentence with timestamps. Nodes without inlinks are extracted as roots and given a timestamp of 1. Nodes linked to by the roots are given a timestamp of 2. For the example in Figure 6(b), s_1 and s_4 are roots with timestamp 1. After running a breadth-first-search, s_2, s_3 and s_5 are tagged with timestamp 2, s_6 with timestamp 3, s_7 with timestamp 4, and s_8 with timestamp 5.

The *Graph Constructor*, *Sentence Ranker* and *Sentence Extractor* following the *Timestamp Tagger* are the same as our previous system (Lin et al., 2007).

3.2 Evaluation Results

The update summarization task provides two document sets for each document cluster. Summarization systems were required to give generic summaries for Set A, and update summaries for Set B on the assumption that users already read documents from Set A.

In total, 58 systems participated in the update summarization task, resulting in a total of 72 submitted system runs. We applied our augmented TSG system on both sets. Table 1 shows our ranking of ROUGE-2 and ROUGE-SU4 for Set A and Set B, as well as an overall ranking AB. The results show that our system did not perform well on Set A. This may be because using temporal information is not helpful in generic summarization, and applying the random walk alone for ranking is not sufficient to extract good summary sentences. However, our system yields a large improvement in ranks when applied to the update summarization task for Set B. This suggests that information extracted from the event ordering is helpful in detecting updates in news reports.

	A	B	AB
ROUBE-2	47	27	41
ROUGE-SU4	43	28	38

Table 1: Ranking of ROUGE-2 and -SU4 over 72 system runs for individual sets A and B, as well as the overall ranking AB.

4 Mining Opinion Snippet Sentences

To recap, the opinion summarization pilot examined the problem of "generat[ing] well-organized, fluent summaries of opinions about specified targets, as found in a set of blog documents", as described by the TAC guidelines. In particular, relevant text snippets from documents were provided by NIST, as output from the corresponding question answering (QA) task. The relevant text snippets can be thought of as pinpointing the location of relevant, opinion-oriented information in each input document, but without identifying the semantics of the opinion.

Queries are asking for summarized subjective answers, and snippets may correspond to document sentences that express opinions. Our processing pipeline aims to recover this information and use it in creating the summary. We capitalize on the provided QA snippets as a rich form of input, and process each query according to the following steps:

1. Assign polarity to the queries;
2. Assign polarity to the provided snippets after suitable expansion;
3. Match snippets to queries, based on polarity and similarity; and
4. Concatenate the expanded snippets sentences for the matched snippets for a query using simple rules.

We first present an offline subjective term mining process that produces pseudo-sentences consisting of terms with high mutual information, which will be applied in snippet sentence expansion, and a polarity classifier for general sentences, which can classify query sentences as well as snippet sentences.

4.1 Offline Term Mining

Previous work (Kobayashi et al., 2007) noted that opinions on particular topics often revolve around topic-specific aspects. For instance, when evaluating an MP3 player, we may discuss its battery life, voice quality, interface quality, etc. Automatically discovering these aspects for a query would thus be useful in subsequent matching.

Consequently, in preparation for the opinion task, we collected a large number of web pages from prominent opinion and review sites (like *Rate It All* [2] and *Product Review Australia* [3]). We hypothesize that the term distribution among opinions about the same topic reflects users' common interests. More specifically, terms from reviews/opinions with the same polarity and on identical topics as the sentence under consideration are most relevant. We extract open-class words, such as nouns, verbs, adverbs and adjectives, for which we calculate mutual information for positive and negative opinions under a topic. Those words with low mutual information are removed from the list. We collected two lists of pseudo-sentences consisting of subjective terms with high mutual information from both *Rate It All* and *Product Review Australia* datasets.

Example: Under the topic of *Digital-Camera*, we retrieved 1416 distinct reviews from *Product Review Australia* that were distributed over 61,842 distinctive terms in the positive, negative and overall sections of the web site. We find high-frequency terms such as *camera, good, use, great, quality, photo, easy, picture, feature, battery, zoom, price*, etc. Furthermore, we collected adjectives such as *inch, lightweight, ease, effective, awesome, powered, superb, fantastic, 10x, super, monitor, simple* that have higher mutual information with positive opinions, whereas adjectives such as *lack, slow, disappoint, noisy, annoy, blurry, poor, flimsy, grainy, working, difficult, expensive, sunlight, hate*, have higher mutual information with negative opinions. Note that some of the collected adjectives are not easily associated with positive or negative opinions in any general sense.

4.2 Classifying Sentence Polarity

To classify an input sentence (which can originate from the query or a snippet) as positive or negative, we first tag the input for parts-of-speech and extract a list of open-class words: nouns, verbs, adverbs and adjectives from it. We then find the polarity scores for these words. The polarity score of the query is the average of its word scores.

Polarity scores for open-class words: We utilize a subjectivity lexicon (Wilson et al., 2005)

[2]http://www.rateitall.com

[3]http://www.productreview.com.au

to find word polarity. If the word's lemma is present in the subjectivity lexicon, its score is assigned based on the prior polarity available in the lexicon, which can be positive, negative, both, or neutral. A score is assigned based on the value of this attribute: positive gets 0.75, negative gets -0.75, both gets 0.25, and neutral gets 0. In addition, if the word is tagged as weakly subjective, the score is further reduced by 0.25 if it is greater than 0, or increased by -0.25 if it is less than 0. In the case where the word does not appear in the lexicon, we fall back and use WordNet to find synonyms of that word. The polarity score assigned is then the average of non-zero polarity scores of the synonyms.

4.3 Classifying Query Polarity

Our analysis on the sample queries revealed that there are common natural language patterns used in the query formulation. This feature could be accounted for by using dependency parsing. Hence, we utilize Minipar (Lin, 1998) to detect patterns with high frequency. Typical patterns include:

- What [pos/neg] reasons ...?
- What reason(s) (are given) for [pos/neg] opinions ...?
- What reason(s) (are given) for [liking/disliking] ...?
- What [pos/neg] properties of (sth) are [favoured/disfavoured]?
- What make(s)/made people [like/dislike] ...?
- Why do people [like/dislike] ...?

Based on our observations of these typical patterns, we designed a set of linguistic rules to check the polarity of a query by examing opinionated words in the square brackets. If no such pattern is found or the score assigned by our rules is less than 0.5, we back off to use the normal polarity classification for a general sentence as described in Subsection 4.2, or a combination of the two if both scores are less than 0.5. Furthermore, if there is any negation indicated by cue words or morphemes (e.g., *not*, *n't*, *hardly*), the polarity will be reversed.

4.4 Expanding and Classifying Snippet Sentences

Given a snippet, our system locates the sentence containing this snippet in the document by selecting the sentence with the highest cosine similarity to the snippet. We term the selected sentence as the *snippet sentence*, and attempt to ascertain its sentiment polarity and match it to a query. A good snippet sentence that is eventually chosen to become a summary sentence should: (1) match a query's topic/focus, and (2) have the same polarity as the query.

From a manual analysis of the development data, we believed that extending the snippet sentence to encompass its context is helpful for polarity classification, especially when the original snippet sentence is short. Our system expands a snippet sentence by considering its previous and following sentences, if they exist. The maximum length of this observation window is set to three, resulting in an expanded snippet context of one to three sentences.

We compute a *patternScore* based on the existence of linguistic constructs like adverbials of cause ("because", "as", "due to the fact that", etc.) and effect ("therefore", "so", "consequently", etc.) discovered through simple string matching. For example, if the snippet sentence starts with the conjuctive adverbial "Therefore" or "As such", etc., then the previous sentence is quite likely to contain the reasons for this sentence and hence will be included in the snippet context. Similarly, the next sentence is selected if it is introduced by a conjunctive adverbial like "Furthermore", "Moreover", or "What's more", etc. The *patternScore* is then assigned in the range of 0 and 1 using some hand-crafted rules. We also calculate *rateItAllScore*, which is the similarity of the sentence and the list of sentiment-carrying pseudo-sentences constructed from the high mutual-information words extracted offline from our *Rate It All* dataset.

Whether the snippet sentence should extend its previous and (or) next sentence is determined by a score s:

$$s = 0.5 * patternScore + 0.5 * rateItAllScore \quad (1)$$

If $s \geq 0.25$ then that context sentence is added. After the expanded snippet is obtained, its polarity is calculated according to the average polarity among these sentences, where the polarity of a single sentence is calculated by the process in Subsection 4.2.

4.5 Assigning Snippets to Queries and Synthesizing Expanded Snippets

Next, we assign snippets to queries with the same polarity. Note that a target may be associated with more than one query. In the case where multiple queries share the same polarity, we need to decide which query a snippet will be paired with. We calculate the similarity between that snippet and all queries with the same polarity sign and the query most similar is paired with that snippet. For each query and its set of matched snippets, we concatenate the corresponding expanded snippet sentences together, in a simple process that accounts for capitalization, punctuation and line breaks. This final output is then returned as the summary for this query.

4.6 Evaluation Results

In total, 19 teams participated in the opinion summarization task and 36 runs were submitted. We submitted two runs: one with ID 2 and the other 20. Run2 uses the selective snippet expansion with Equation 1, whereas Run20 always includes the two contextual sentences. The result shown in Table 2 indicates that Run2 (ranked fourth in pyramid F-score and third in responsiveness) yields better performance. When the context is selectively included, not only that the snippet's polarity is more accurately assigned (so that the pairing of snippets and their queries are more correct), but the content of the summary is also more grammatical and coherent, as irrelevant sentences have been removed. We feel this result is intuitive as in a blog environment, arguments are usually expressed in a compact form so that they do not span over many sentences. In a nutshell, we believe that for this opinion summarization task, it is importance to construct snippets' context precisely and in doing so, it is better to be conservative rather than liberal.

	Run2	Run20
Used answer snippets	Yes	Yes
Pyramid F-score (Rank)	0.461 (4)	0.329 (10)
Grammaticality	4.727	4.455
Non-redundancy	5.455	4.909
Structure/Coherence	2.955	2.5
Fluency/Readability	3.909	3.273
Responsiveness (Rank)	5.318 (3)	4.909 (6)

Table 2: Evaluation results for our two runs.

5 Conclusion

This year our team participated in both TAC 08 summarization tasks. For update summarization, we investigated the impact of incorporating temporal relations among fine-grained events, where we employed Tarsqi toolkit for this purpose. Results are promising: the temporal information extracted by the toolkit were used in the summarization process and much better ROUGE rankings were achieved. This suggests that temporal information can help to identify novel content in news articles and it is feasible to extract such information by automatic systems. We plan to conduct further tests to ascertain whether the gains are systematically due to incorporating such event information.

Opinion summarization starts with opinion recognition. Apart from that, we hypothesized that the contexts of mentions of opinions also play an important role in forming a useful opinion summary. This hypothesis is supported as our baseline opinion summarization system achieved competitive performance – it simply collects sentences that surround mentions of opinions as their contexts. We also showed that terms reflecting people's interests and certain linguistic patterns are good features to select contexts that are correlated with improved summaries. We believe that the associated context detection techniques are potentially applicable to other NLP problems such as Question Answering (QA) as well, deserving further research efforts.

Although the settings of this year's summarization tasks strongly encourage abstractive summarization, we have chosen instead to focus on pushing sentence extraction methods further, in the hopes that our methods will also work at the sub-sentence level. In future work, we plan on building such Natural Language Generation (NLG) capabilities into our future TAC summarization systems.

6 Acknowledgments

Our team would like to acknowledge the help of Xuan Wang in helping us get started with the blog summarization task.

References

Terry Copeck, Diana Inkpen, Anna Kazantseva, Alistair Kennedy, Darren Kipp, and Stan Szpakowicz. 2007. Catch What You Can. In *Proceedings of DUC 2007*.

DUC. 2007. DUC 2007 Guidelines (www-nlpir.nist.gov/projects/duc/guidelines/2007.html).

Brigitte Endres-Niggemeyer. 1998. *Summarizing information*. Springer New York, New York.

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

Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1065–1074, Prague, Czech Republic, June. A.

Maheedhar Kolla, Olga Vechtomova, and Charles L.A Clarke. 2007. Comparison of Models Based on Summaries or Documents Towards Extraction of Update Summaries. In *Proceedings of DUC 2007*.

Elizabeth DuRoss Liddy. 1991. The discourse-level structure of empirical abstracts: an exploratory study. *Inf. Process. Manage.*, 27(1):55–81.

Ziheng Lin, Tat-Seng Chua, Min-Yen Kan, Wee Sun Lee, Long Qiu, and Shiren Ye. 2007. NUS at DUC 2007: Using Evolutionary Models of Text. In *In Ducument Understanding Conference (DUC) notebook papers*, Rochester, NY, USA, April.

Dekang Lin. 1998. Dependency-based Evaluation of MINIPAR. In *Proc. Workshop on the Evaluation of Parsing Systems*, Granada.

Inderjeet Mani and George Wilson. 2000. Robust temporal processing of news. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 69–76, Morristown, NJ, USA. Association for Computational Linguistics.

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. 2003. Inferring temporal ordering of events in news. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 55–57, Morristown, NJ, USA. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July.

Prasad Pingali, Rahul K, and Vasudeva Varma. 2007. IIIT Hyderabad at DUC 2007. In *Proceedings of DUC 2007*.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. Evita: a robust event recognizer for qa systems. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 700–707, Morristown, NJ, USA. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.

Shiren Ye, Long Qiu, Tat-Seng Chua, and Min-Yen Kan. 2005. NUS at DUC 2005: Understanding documents via concepts links. In *Proceedings of DUC 2005*.