TLR at DUC 2006: approximate tree similarity and a new evaluation regime

Frank Schilder
R&D
Thomson Legal & Regulatory
610 Opperman Drive
Eagan MN 55123, USA
Frank.Schilder@Thomson.com

Bridget Thomson McInnes
Department of Computer Science
and Engineering
University of Minnesota
200 Union Street SE
Minneapolis MN 55455, USA
bthomson@cs.umn.edu

Abstract

We propose modifications to a summarization system that is based on computing the tree edit distance between dependency parse trees of reformulated questions and candidate sentences. We modify a recently introduced approximate tree edit distance metric by using mutual information between stemmed words for similarity matching of sub-trees. We also propose an approximate way of deriving anaphoric expressions from the topic description and finally leverage the subject-object structure of the question. In addition, we discuss the current evaluation regime and propose changes to the evaluation and the summarization task.

1 Introduction

Our summarization system uses an approximate tree similarity metric (Augsten et al., 2005) computed for the dependency trees of reformulated questions and candidate sentences from the topic clusters. Building on a similar system from last year we focused this year on (a) improving the matching of subtrees (b) addressing the problem of the highly elliptic questions to be answered for this task, and (c) leveraging the subject-object structure of the questions for improving the overall score of our system.

The solutions we developed are the following:

(a) When computing the similarity score for the original tree similarity metrics, a high penalty was given if two nodes are not the same. Synonyms or hypernyms should get no or a very low penalty instead. We used a large news corpus and computed proximity scores similar to Lin (1998). We now give a bonus if two words have a high mutual information score.

(b) Questions for this task were sometimes very elliptical and did not contain any meaning-carrying words that would ensure a high score for the tree similarity metric. Instead of using a definite description resolver, we computed a similarity score between the parsed topic phrases (e.g., *wetlands value and protection*) and the parse trees of the candidate sentences in addition to the scores between parses of the reformulated question and the sentences.

(c) After analyzing the questions used for this task, we noticed that the actual topic of the question is most often the subject of the question. Based on that observation, we boosted the scores for words that were the subject of the question as well as the candidate sentence (e.g. Q: *Which countries...* A: *Germany...*).

In addition to discussing these new features employed by our summarization system, we would like to point to several shortcomings of the current evaluation regime and propose a modification of the summarization task for next year.

Most prominent is the observation that the automatic evaluation metrics do not reflect the still existing gap between automatic summaries and human-written summaries. Among other things, it seems that the overall criteria of readability which is influenced by coherence and referential clarity do not seem to be taken into account in the way automatic metrics such as ROUGE (Lin, 2004) or BE (Hovy et al., 2005) are computed.

Instead, we would like to propose a more user-focused evaluation that will allow us to drive the research in summarization by focusing also on overall coherence and understandability of the automatically generated summaries. An extrinsic study needs to be carried out to investigate features such as the time a user needs for reading the summary until making a judgment regarding their usefulness. Another modification of the current evaluation regime would be the generation of human-based

summaries that are less perfect than the models we compare against currently. Randomly scrambling summaries would, for example, lead to summaries with the same information content but with a defective discourse structure. Another type of a less-than-perfect summary could be generated by partly merging them with automatically generated summaries.

Finally, we would like to propose a modified summarization task for next year. The query-based multi-document summarization task should be modified in the following way: a small restricted set of questions that focus on opinions or/and sentiments expressed on a particular topic.

2 Our approach

Our approach is based on computing similarity scores between questions and candidate sentences. In particular, our measure computes a distance between the dependency parse trees of two sentences. Normally, tree distance approaches are based on the tree similarity metrics developed by (Shasha and Zhang, 1989). In NLP applications, this approach has been used for Question Answering (Punyakanok et al., 2004), and Textual Entailment (Kouylekov and Magnini, 2005; Marsi et al., 2006).

For this year's system, we applied a recently developed approximate tree matching algorithm (Augsten et al., 2005). This approach has a number of advantages over the original tree edit distance algorithm developed by (Shasha and Zhang, 1989). Differences in the actual tree structure become more pronounced and it is computationally far less expensive.

In general, our summarization algorithm performs the following four major processing steps:

- **Linguistic smoothing (LS).** This year, we only used our pronoun resolution module that is built on top of LingPipe.[1]. Due to time and resource constraints no other techniques (e.g. coherence, temporal information) were used this time.
- **Question reformulation (QR).** Some preprocessing of the questions is necessary so that candidate sentences form the document collections can be compared to the query to compute a distance score. The second processing step translates every question into an affirmative sentence (e.g. *What is the World Bank? → The World Bank is *ANSWER**.). We modified this module this year so that *What NP*-questions would not loose the NP in the reformulated **ANSWER***-statement (e.g. *Which countries... →Countries...*). In the remainder of the paper, we will refer to these sentences as the **ANSWER***-statements.
- **Filtering (FI)**. Sentences were filtered in order to reduce the subsequent processing load.
- **Tree extraction (TE).** Using the dependency parser MiniPar (Lin, 1998b), we parsed the reformulated questions and all the sentences in the topic collection. A tree similarity score was computed between the **ANSWER***-statements and the sentences from the collections. The most similar sentences, with respect to the given question, were extracted from the set of candidate sentences in the collection.

This year, we used an approximate tree similarity metrics (Augsten et al., 2005) and also experimented with the way the overall score for a candidate sentence was computed. The following sections go into more detail on how the QR and TE components were changed this year.

2.1 Linguistic smoothing

We only employed our pronoun-substitution module from last year in order to avoid dangling pronouns in extracted answer sentences.

2.2 Question reformulation

The Question Reformulation (QR) tool transforms a set of questions into an affirmative or *ANSWER*-statement for each question. For example, the *ANSWER*-statement for the input question *Why are wetlands important?* is as follows:

(1) Wetlands are important because *ANSWER*.

We then use the **ANSWER** tag to indicate a place for the actual answer to be incorporated directly into the statement. We compare the reformulated sentence to the sentences from the document collection and similar ones are extracted as summary candidates.

The QR tool contains four modules: a sentence splitter, a part-of-speech (POS) tagger, a shallow parser, and the statement generator. The sentence splitter and POS tagger process the data so that it can be parsed by the shallow parser. First, the sentence splitter extracts the questions and then the POS tagger tags each word in the sentence. The system uses the BRILL POS tagger (Brill, 1992).

The third module is the shallow parser. The parser used for this system is the CASS parser developed by Abney (1990). CASS consists of a series of cascading finite-state transducers. In addition to the partial parse of the sentence, the parser also provides subject and object information.

The last module, the statement generator, extracts four components from the CASS parser output: the question word, the main verb, the subject, and the object. It then feeds the sentence into a cascading set of rules that, based on these four components, creates a template statement that the question can be transformed into.

[1] `http://alias-i.com/lingpipe/`

Consider the following example query, *Where are they threatened?*. Given the information from the CASS parser the statement generator sequentially applies the following rules: the first rule set identifies the question word. In this case, the sentence contains the question word *where*. The second rule set identifies that the verb is (*are threatened*.) Finally, the subject and object are identified. In our example, the subject is *they* and an object does not exist. The sentence is then transformed using the following answer template:

(2) <subject> <verb> in *ANSWER*

This template generates the following sentence for our example:

(3) They are threatened in *ANSWER*

2.3 Filtering

The filtering of the sentences is done with simple word overlap between question and candidate sentences and a approximate matching of words. We used collocation information mined from newspaper articles similar to (Lin, 1998a). By computing the mutual information between the words, we require at least 3 words in the question similar to words in the candidate sentences to be selected for the final step. The computation of the mutual information between words is described in more detail in the following section.

2.4 Tree extraction

2.4.1 Tree similarity metrics

In (Schilder et al., 2005), we showed how a tree similarity metric can be used for the DUC 2005 task. The original tree edit distance algorithm allows for three different operations in order to transform a tree t into a tree h: (a) delete, (b) insert and (c) change. Depending on the application, one challenge lies in choosing the right penalties for these three operations. In particular, the change operation should be carefully chosen in order to penalize for example synonyms. In addition, the metric should also allow for capturing structural similarity.

The approximate tree similarity metric was proposed by (Augsten et al., 2005). For the remainder of this paper, this metrics will be called *adist*. The tree structures used for this similarity metric was derived from dependency parses we obtained from running the MiniPar parser on all sentences (Lin, 1998b).

2.4.2 *adist*

The algorithm proposed by (Augsten et al., 2005) uses so-called p, q-grams which are computed on the basis of an extended tree $\mathrm{T}^{p,q}$ that contains extra empty nodes (see figure 1). The p, q-grams are derived from all possible subtrees of $\mathrm{T}^{p,q}$ given an anchor node that has $p-1$ ancestors and q children. In the extended tree in figure 1, the node *bought* would be a possible anchor that has $*$ as ancestor and combined with the bag of all possible q children produces the following p, q-grams:
$\{[*, bought, *, *, John], [*, bought, *, John, car],$
$[*, bought, John, car, *], [*, bought, car, *, *]\}$.
By generating all possible p, q-grams for two trees we obtain a profile $\mathrm{P}^{p,q}(t)$ for a given tree t. Given a profile for each tree, the similarity score for two trees t_1, t_2 is computed as follows:

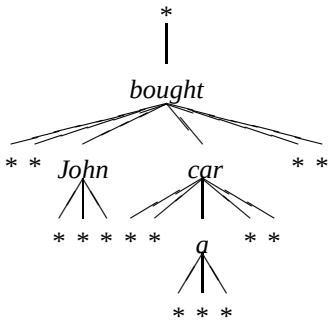
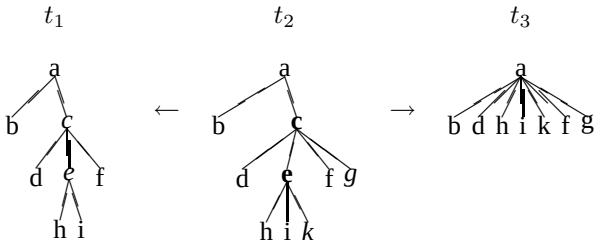


Figure 1: p, q-extended tree with $p = 2, q = 3$

$$\Delta^{p,q}(t_1, t_2) = 1 - 2\frac{\mid \mathrm{P}^{p,q}(t_1) \cap \mathrm{P}^{p,q}(t_2) \mid}{\mid \mathrm{P}^{p,q}(t_1) \cup \mathrm{P}^{p,q}(t_2) \mid} \quad (4)$$

Figure 2 shows how the tree edit distance (i.e., $\mathrm{dist}_{\mathrm{ed}}$) and the approximate edit distance metric (i.e., $\Delta^{p,q}$) differ when structural similarities are of importance. Tree t_1 is structurally still quite similar to t_2, but t_3 differs substantially from t_2 because parent nodes were deleted. Note that the result for $dist_{ed}$ is still the same for these two trees, whereas $\Delta^{p,q}$ captures the structural difference much better.



$\mathrm{dist}_{\mathrm{ed}} = 2$ $\quad$ $\mathrm{dist}_{\mathrm{ed}} = 2$
$\Delta^{2,3} = 0.32$ $\quad$ $\Delta^{2,3} = 0.89$

Figure 2: Structural information is captured better with *adist*

3 Improvements

3.1 Finding similar p, q-grams

The original adist algorithm only checks for the intersection of p, q-grams . It does not cover close matches, synonyms or hyponyms. This becomes particularly important when the reformulated question contains a noun such as *countries*. A potential answer sentence would probably not contain the word *countries* but instead actual countries such as *Germany, Japan* etc. Consequently, synonyms or hypernyms in a p, q-gram should get a bonus when matched against the entire p, q-grams . We used a large news corpus and computed proximity scores similar to (Lin, 1998a). Based on these proximity scores, a bonus between two words is computed derived from the mutual information for these words.

Mutual information is a measure for the mutual dependence of two variables (i.e., the information that is shared between the two variables). (Turney, 2001) successfully used this measure to determine synonyms. For our application the mutual information score ensures that we give word pairs a bonus when they share information (e.g. synonyms), but filter them out when the shared information is minimal:

$$\mathrm{mi}(x, y) = log_2 \left(\frac{p(x, y)}{p(x) * p(y)} \right) \quad (5)$$

In order to compute a score for an p, q-gram, we need to compare the entire tuple. Let pq_1 and pq_2 be the p, q-grams we want to compare. Each word in this p, q-gram can be accessed by [n]. $\mathrm{pq}_1[3]$, for example, is the word *bought* in the p, q-gram $[*, countries, bought, *, *]$. In order to compute a score for the entire p, q-gram , we look for the highest mutual information score in the pairwise comparison of the p, q-grams entries. If only one of mi score, however, is less than 0, the entire comparison score is 0.

$$s(\mathrm{pq}_1, \mathrm{pq}_2) = \begin{cases} \arg\max_{n \in \{1..|p+q|\}} \mathrm{mi}(\mathrm{pq}_1[n], \mathrm{pq}_2[n]) \\ \qquad \text{if } \forall n\ \mathrm{mi}(\mathrm{pq}_1[n], \mathrm{pq}_2[n]) \geq 0 \\ 0 \\ \qquad \text{otherwise} \end{cases}$$

This computation re-introduces a complex comparison operation that leads to a slower computation of the entire tree similarity score.

3.2 Amending elliptical questions

The success of our approach obviously hinges on the quality of the reformulated question. In particular, if there were many meaning-bearing words in the question, the tree matching algorithm should come up with a good answer sentence. Unfortunately, questions were sometimes very elliptical (e.g. *Name those involved, if possible*).

Carrying out pronoun resolution and in particular resolving definite descriptions seems to be a crucial prerequisite in order to successfully answer the question given in this task. However, instead of automatically resolving anaphoric expressions and elliptic constructions, we leveraged the topic description. Each topic came with a short high level description (e.g. *crime and law enforcement in China*). We computed the $1, 3$-gram similarity between these phrases and the candidate sentences. Note that a $1, 3$-gram covers basically tree of depth 1 which would be NPs or modifiers.

In order to factor in this similarity score, we computed the linear combination of the two scores, but future experiments will entail different combinations.

3.2.1 Leveraging the subject-object structure

The question's subject often specifies the type of 'what' is asked for (e.g. *What countries...*). We decided to exploit the subject-object structure derived from the dependency parse. The tree similarity metrics should get a higher score if the subject of the questions also matches with the subject of the **ANSWER**-statement.

Based on that observation, we boosted the scores for words that were the subject of the question as well as the candidate sentence (e.g. Q: *Which countries...* A: *Germany...*). After testing different parameters we multiplied the s-score for each $\mathrm{pq}[n]$ by 2, if both words were in the subject position.

4 Results

We were able to improve our results over last year's results and reach competitive results in the automatic metrics as well as the human-annotated ones. For the linguistic quality, however, we only received average results (cf. table 1).

metrics	system ø	our score	our rank
ROUGE-2	0.0739	0.0858	7
ROUGE-SU4	0.1293	0.1438	8
BE-2	0.0364	0.0456	8
Ling. quality	3.3819	3.38	17
Response-content	2.5423	2.89	6
Response-overall	2.1874	2.6	3

Table 1: Average system scores and our scores and ranks (out of 34 systems)

Interestingly enough, we received a relatively high score for Responsiveness-overall. This score measures the amount of information that is expressed in the summary to answer the questions and the overall readability. Although we had only an average score in linguistic quality, we scored better in this combined score than in the

Responsiveness-content only score. This seems to indicate that the linguistic quality score does not make good predictions on readability of the summary.

Further observations can be made when viewing the plot of the ROUGE-2 results against the Responsiveness-overall scores. Given the combination of these two scores there is only one system (i.e. system 23) that scored higher than our system. The best system in responsiveness (i.e. 27) has a relatively low ROUGE-2 score (0.08082; rank 12). The best ROUGE-2 system (i.e system 24), on the other hand, has relatively low responsiveness scores (content:2.88;overall: 2.44) and is ranked fifth for each of these scores.

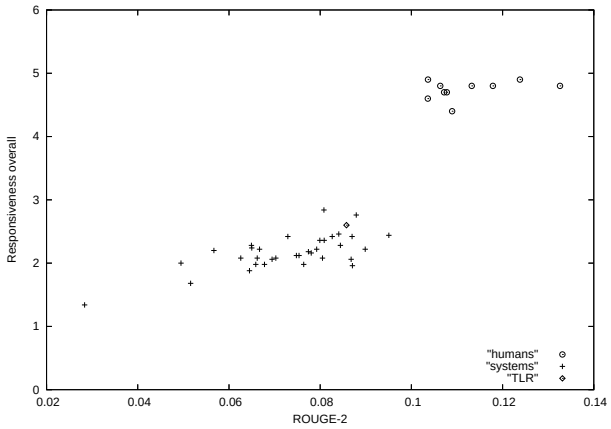


Figure 3: The correlation between ROUGE-2 and Responsiveness overall score for all models and peers

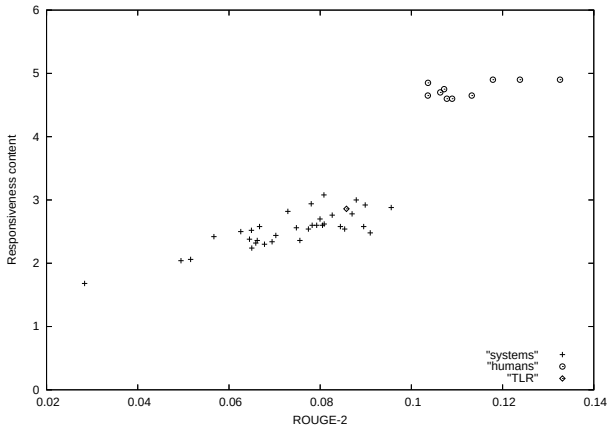


Figure 4: The correlation between ROUGE-2 and Responsiveness content score for all models and peers

5 A new evaluation regime

The ultimate goal of the DUC roadmap for evaluation of query-based multi-document summaries seemed to be a user-focused evaluation metric. We are still quite far away from this goal. First of all, current evaluations (human-based and automatic) do not take into account whether a user does indeed benefit from reading a summary. The benefit of reading a summary may also depend on the actual task the user has to carry out. Consequently, there is no real user-oriented evaluation if we do not know the task a user has to carry out with the help of the summary.

Secondly, it seems that the automatic evaluation methods do not reflect the gap between automatically generated and human written summaries. There is still a significant difference between the best system and the worst human-written summary score (2.84 vs. 4.4), whereas the ROUGE-2 scores seem to suggest that this gap is closing (0.09558 vs. 0.10361). Moreover, the automatic measures seem to indicate the there is a wide quality difference in the human-written summaries, even though if you look at the responsiveness scores there is none.

Lastly, the responsiveness score is not very fine-grained and is not useful when addressing the weaknesses of a system. Instead, we should develop a scoring system that provides a more detailed picture of why a summary received its overall score.

5.1 Extrinsic studies

It has already been proposed to carry out an extrinsic evaluation for DUC last year (Daumé III and Marcu, 2005). We can not emphasize enough that such a study would probably be very helpful, because it will bring the user back into the picture. One important requirement for an extrinsic study is the definition of a task. Such a task could test the summaries for completeness or whether they are a good indicator of the content of the full article. The current responsiveness evaluation seems only to test whether the summary is informative (i.e., does the summary cover all the required information). An indicative summary, on the other hand, would give a user a tool for deciding which documents she would like to select and read in more depth. We believe that the second task is a more realistic one, because users probably do not want to rely solely on an automatically generated summary, if they are looking for answers for such complex information needs as described in the topics of the current DUC task.

We carried out a task-oriented study with users who needed to decide whether documents returned from a search results are relevant. We choose to implement a two-step system that offered a short teaser sentence to the user. Such a sentence should be indicative of what the actual document is about. However, after clicking on the link to the document, the user had a second chance when she saw a two-paragraph summary of the document that allow her to decide whether the content of the document actually would provide answers to the information need

expressed by the search query. Such as two-step extrinsic evaluation would measure the indicative and informative value of automatically generated summaries.

Finally, a task-oriented evaluation would give a temporal dimension to the evaluation. If the task is to come up with an answer to a question quickly, the time a user needs to find this answer with the help of the summaries and the document collection could be measured.

5.2 Model modification

The responsiveness scores still show a considerable gap between the best system and the worst human-written summaries. It would be useful to investigate how this gap could be filled with not-so-perfect model summaries. A baseline that takes the model summaries and introduce errors could provide data on how a summary degrades in the scoring. One possible modification is the random scrambling of the sentences in the summary. The content would be the same, but readability would be worse. Another baseline system could introduce pronouns by deleting proper nouns.

5.3 Responsiveness, coherence and penalties

Instead of providing a coarse score ranging from 1-5 on responsiveness of the summary, evaluators should have the possibility to give more fine-grained scores. First of all, the sequence of sentences should be matched to the queries given. A summary that contains the information asked for but in a confusing sequence would most likely not be judged very useful by a user. A scoring system needs to be established that gives high scores to correct answer sentences at the beginning of the summary, if they can satisfy the information need for the first question.

Second, the coherence should not only be linked to the question, but also to how well sentences are connected. Evaluators could rate the cohesion between sentence in the summary. By linking the responsiveness with the overall coherence and the local cohesion, scoring would become more transparent.

Moreover, a penalty score should be introduced for sentences that either contradict other statements in the summary or lead to wrong conclusions.

6 A new summarization task

The current query-based summarization tasks has a wide variety of different types of questions. Reducing this variety may lead to summaries that are better tailored towards the information needs of this particular question type. The DUC 2004 competition, for example, focused on *Who is X* question. If we are serious about improving overall discourse structure and referential clarity, we need to come back to a similar task that restricts the number of different questions types.

Only if there is a predictable set of question types, groups will start trying to improve their summaries accordingly. The current setting does not encourage developing different summarization strategies tuned to the different question types.

We propose the following question type that is not as restricted as the *Who is X* question type but would allow to develop techniques that could advance the field in generating more coherent summaries: given a set of 25-50 articles on a controversial topic, summaries are to be generated that present the views of different organizations, people, countries etc. The answers wouldn't be topic, but sentiment-centered. The summarization system would have to develop the following capabilities:

- Determine the views of a person, organization etc.
- Cluster similar views, emphasize differences.

The following two sentences, for example, express two differing views on Guantanamo Bay. IN order to extract these views it's necessary to identify not only the organization names but also respective spokesperson (i.e.' *Tony Snow*).

(6) A United Nations committee report released Friday condemned U.S. treatment of suspected terrorists and called for the closing of the prison camp at Guantanamo Bay, Cuba.

(7) White House spokesman Tony Snow noted [...] that the treatment of detainees there "is fully within the boundaries of American law."

In addition to sentences that express the different views on a topic, it will also be useful to extract facts that are at least partly undisputed and provide background information on the topic.

(8) Since 2002, the U.S. incarcerated 800 captives suspected of terrorist links to al-Qaeda at Gitmo.

7 Conclusions

Our system for this year's task showed improvements in three different areas: (a) using an approximate tree similarity metrics that takes the tree structure into account as well as the similarity between words (b) utilizing the topic description allowed us to simulate the resolution of anaphoric expressions and elliptic constructions, and (c) the subject-object structure was leveraged for the matching of reformulated questions and candidate sentences.

We also propose a change in the evaluation regime and propose a new summarization task: sentiment-centered multi-document summarization.

Acknowledgments

We would like to thanks Andrew McCulloh, Alex Zhou and Tonya Custis with preparing and cleaning the training and test data.

References

S. Abney. 1990. Rapid incremental parsing with repair. In *Proceedings of the 6th New OED Conference*, Waterloo, Ontario.

Nikolaus Augsten, Michael H. Böhlen, and Johann Gamper. 2005. Approximate matching of hierarchical data using pq-grams. In *Proceedings of Very Large Data Bases (VLDB) Conference*, pages 301–312. ACM Press.

Eric Brill. 1992. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Langu age Processing*, pages 152–155, Trento, IT.

Hal Daumé III and Daniel Marcu. 2005. Bayesian summarization at duc and a suggestion for extrinsic evaluation. In *Proceedings of the Document Understanding Conference (DUC)*, Vancouver, B.C., Canada, October 9–10.

Eduard Hovy, Chin-Yew Lin, and Liang Zhou. 2005. Evaluating duc 2005 using basic elements. In *Proceedings of Document Understanding Conference (DUC 2005)*, Vancouver, B.C. Canada, October.

Milen Kouylekov and Bernardo Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In Oren Glickman Ido Dagan and Bernardo Magnini, editors, *Proceedings of the PASCAL Recognising Textual Entailment Challenge*, April.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98*, pages 768–774, Montreal.

Dekang Lin. 1998b. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*, Granada, Spain, May.

Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain.

Erwin Marsi, Emiel Krahmer, Wauter Bosma, and Mariet Theune. 2006. Normalized alignment of dependency trees for detecting textual entailment. In *Second PASCAL Recognizing Textual Entailment Challenge*, Venice, Italy, April.

V. Punyakanok, D. Roth, and W. Yih. 2004. Mapping dependencies trees: An application to question answering. In *Proceedings of AI&Math 2004*.

Frank Schilder, Andrew McCulloh, Bridget Thomson McInnes, and Alex Zhou. 2005. TLR at DUC: tree similarity. In *Proceedings of the Document Understanding Conference (DUC)*, Vancouver, B.C., Canada, October 9–10.

D. Shasha and K. Zhang. 1989. Fast parallel algorithms for the unit cost editing distance between trees. In *SPAA '89: Proceedings of the first annual ACM symposium on Parallel algorithms and architectures*, pages 117–126, New York, NY, USA. ACM Press.

Peter D. Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pages 491–502, Freiburg, Germany.