In *Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management* (CIKM), Kansas City, Kansas, Nov 2-6, 1999.

Training a Selection Function for Extraction

Chin-Yew Lin
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292, USA
Tel: 1-310-822-1511
Email: cyl@isi.edu

ABSTRACT

In this paper we compare performance of several heuristics in generating informative generic/query-oriented extracts for newspaper articles in order to learn how topic prominence affects the performance of each heuristic. We study how different query types can affect the performance of each heuristic and discuss the possibility of using machine learning algorithms to automatically learn good combination functions to combine several heuristics. We also briefly describe the design, implementation, and performance of a multilingual text summarization system SUMMARIST.

Keywords

Automated text summarization, topic extraction, summary evaluation.

1. INTRODUCTION

Automated text summarization systems [39] that generate short and concise descriptions of the essential content of documents have been a dream since late 1950's [23]. The growing amount of online text on the World Wide Web manifests the need of automated text summarization technology today [2, 7, 8, 11, 30, 34, 35, 37, 41]. However, generating a high quality summary requires natural language processing techniques such as semantic parsing, discourse analysis, world knowledge inference, and language generation that are still under research. As a result, most of the current automated text summarization systems produce extracts instead of abstracts. An extract is a collection of important sentences of a document, reproduced verbatim. The importance of a sentence within a document is determined by using various heuristics such as position importance [4], cue phrases [15, 17, 43], signature words [20], word/phrase frequency [9, 23], lexical cohesion [3, 25], discourse structures [27, 28], and the presence of certain word types [1, 15, 17]. For a brief review of most of these heuristics please refer to [32].

After a system applies some or all the above heuristics, each sentence has been assigned several different scores. Some method is required to combine these scores into a single score, so sentences can be ranked according to their topic-bearing degree. However, it is not immediately clear how the various scores should be combined for the best result-nor even if some of the scores should be left out at all. Various approaches have been described in the literature. Most of them employ some sort of combination function, in which coefficients assign various weights to the individual scores, which are then summed. Kupiec et al. [17] and Aone et al. [1] employ Bayesian classifiers to derive coefficients for their systems. Mani & Bloedorn [25] use Standard Canonical Discriminant Function [40], C4.5-Rules [33], and AQ15c [45]. Marcu [28] experimented with GSAT-like algorithm [38].

In this paper we compare the performance of various heuristics for generating informative generic/query-oriented extracts for newspaper articles in order to learn how topic prominence can affect the performance of each heuristic. We study how different query types can affect the performance of each heuristic and discuss the possibility of using machine learning algorithms to automatically learn good combination functions. We describe an automated text summarization system SUMMARIST [14] in Section 2. In Section 3 we describe the heuristics used to identify topic-bearing sentences, testing corpus, evaluation setup, and the results of the evaluation. We conclude with main findings and future directions.

2. SUMMARIST

The goal of SUMMARIST is to generate summaries of multilingual input texts. SUMMARIST can process English, Arabic, Bahasa Indonesia, Japanese, Korean, and Spanish texts at this time. SUMMARIST combines existing robust natural language processing methods (morphological transformation and part-of-speech tagging), symbolic world knowledge (WordNet Miller et al. [31], and dictionaries), and information retrieval techniques (word counting and term distribution) to achieve high robustness and better concept-level generalization.

The core of SUMMARIST is based on the following 'equation': *summarization = topic identification + topic interpretation + generation*. These three stages are:

Topic Identification: Identify the most important (central) topics of the texts [21]. SUMMARIST uses positional importance [9, 21], cue phrases [9, 32, 43], and term frequency. Importance based on discourse structure will be added later [27, 28]. This is the most developed stage in SUMMARIST.

In *Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management* (CIKM), Kansas City, Kansas, Nov 2-6, 1999.

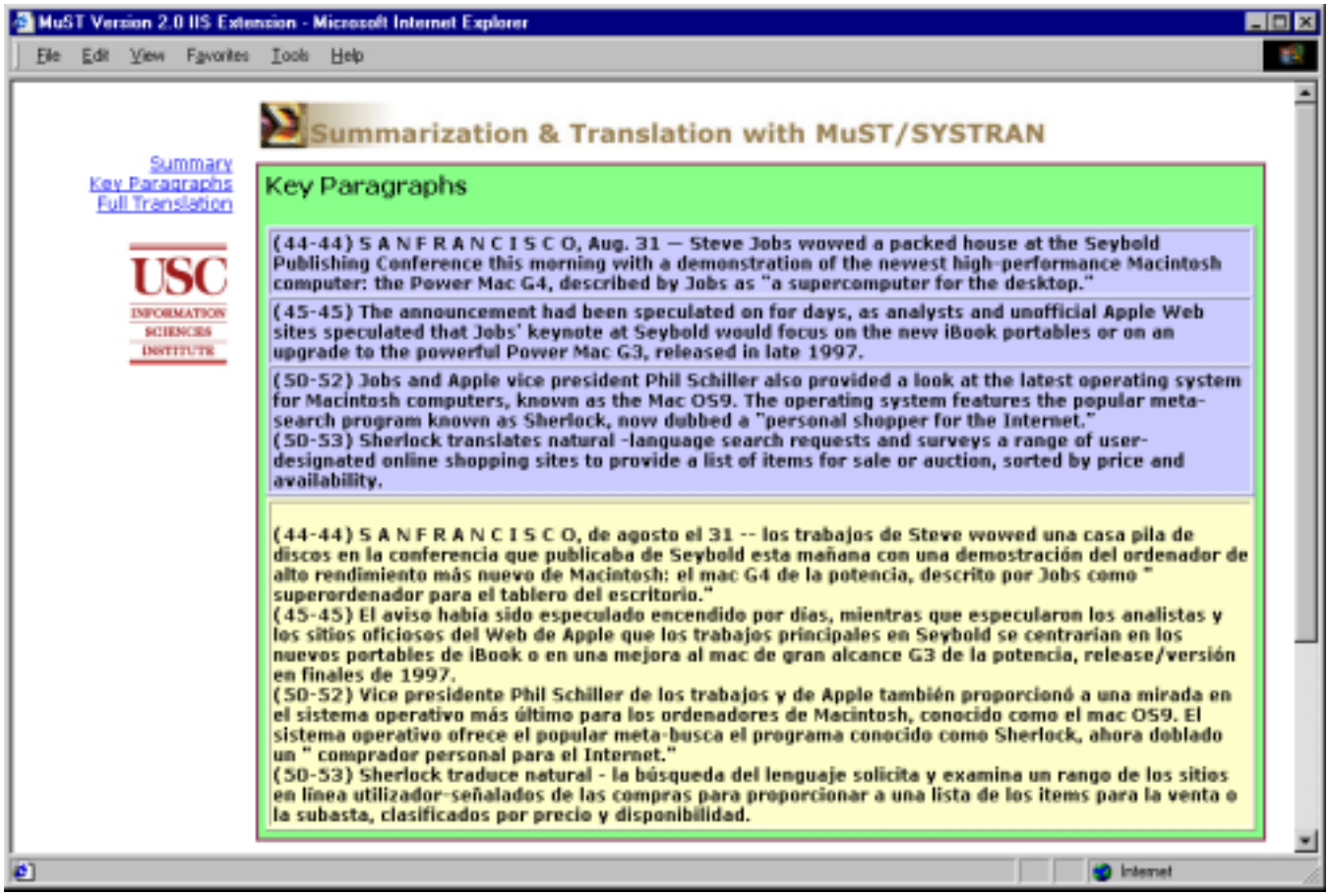

Figure 1. Web news page summary and translation by the MuST system. The upper panel is the English summary generated by SUMMARIST and the lower is its Spanish translation generated by Systran online machine translation software.

Topic Interpretation: To fuse concepts such as waiter, menu, and food into one generalized concept restaurant, we need more than the simple word aggregation used in traditional information retrieval. We have investigated concept counting [19] and topic signatures [20] to tackle the fusion problem.

Summary Generation: SUMMARIST will be able to generate summaries in various formats such as keywords (important noun phrases), extracts (important sentences in original texts), template-based summaries [29] (generated from pre-specified templates), and refined summaries (generated by a sentence planner and realizer) [13, 18]. However, our current system can only produce keyword and extract type summaries.

Figure 1 shows MuST [22], a multilingual text summarization and translation system that embeds SUMMARIST, summarizing and translating a ABC News page[1]

In the following, we briefly describe different heuristics used in SUMMARIST's Topic Identification stage to score terms and sentences and then detail the effectiveness of each heuristic through in-depth evaluation. The score of a sentence is simply the sum of all the scores of content-bearing terms in the sentence. These heuristics are implemented in separate modules using inputs from preprocessing modules such as tokenizer, part-of-speech tagger [6], morphological analyzer, term frequency and *tf·idf* weights calculator, sentence length calculator, and sentence location identifier.

3. COMPARING the EFFECTIVENESS of HEURISTICS

Initially, we implemented for SUMMARIST a straightforward linear combination function, in which we specified the coefficients manually, by experimentation. This hand tuning method does not guarantee consistent performance over a large collection. As we found in the formal TIPSTER-SUMMAC evaluation of various summarization systems, organized by DARPA [10], the results of this function were decidedly non-optimal! Since consistent performance and graceful degradation are very important for any automated text summarization system, a better understanding of the relative strength of each heuristic in different task setting, the stability of each heuristic, how and when to combine various heuristics are needed.

Given the complexities inherent in summary evaluation, one may decide to simplify the problem a little, by studying the behavior

[1]URL for the original web page is at http://abcnews.go.com under "sections/tech/DailyNews/seyboldapple990831.html". Translation is provided by Systran, http://www.systransoft.com

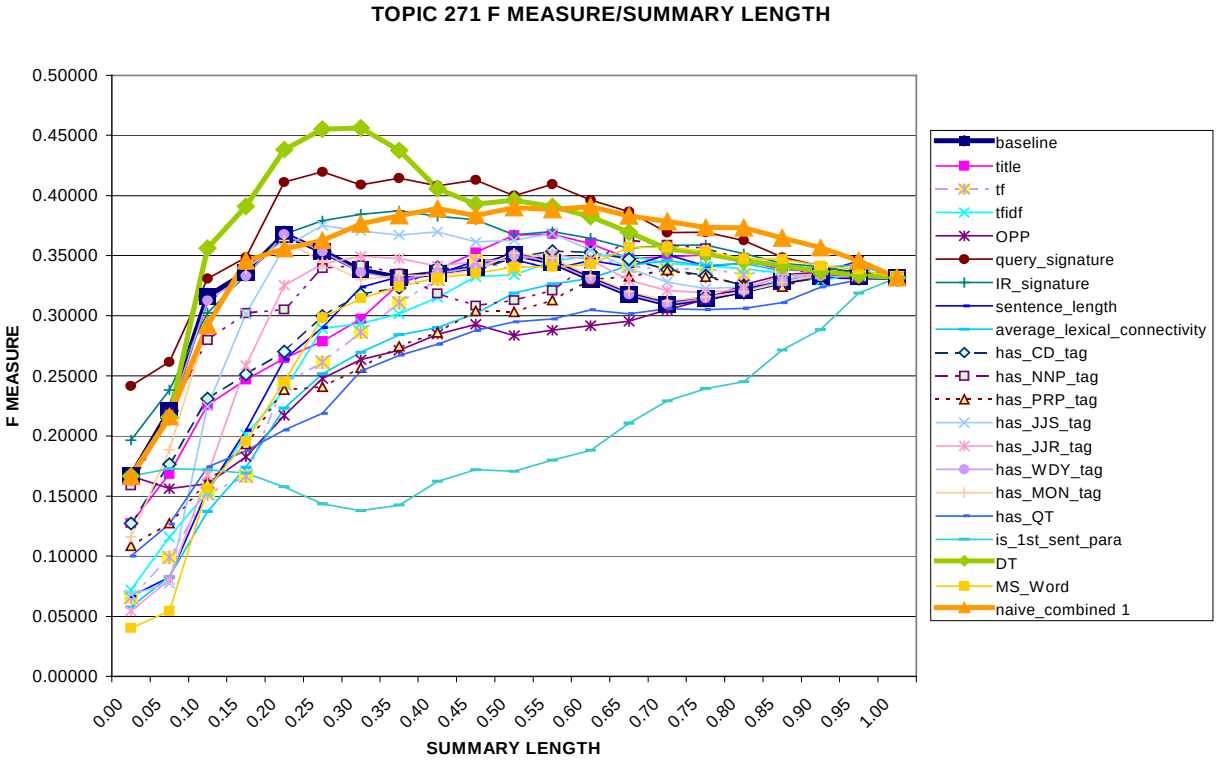

Figure 2(a). F-scores for all parameters and features, Topic 271.

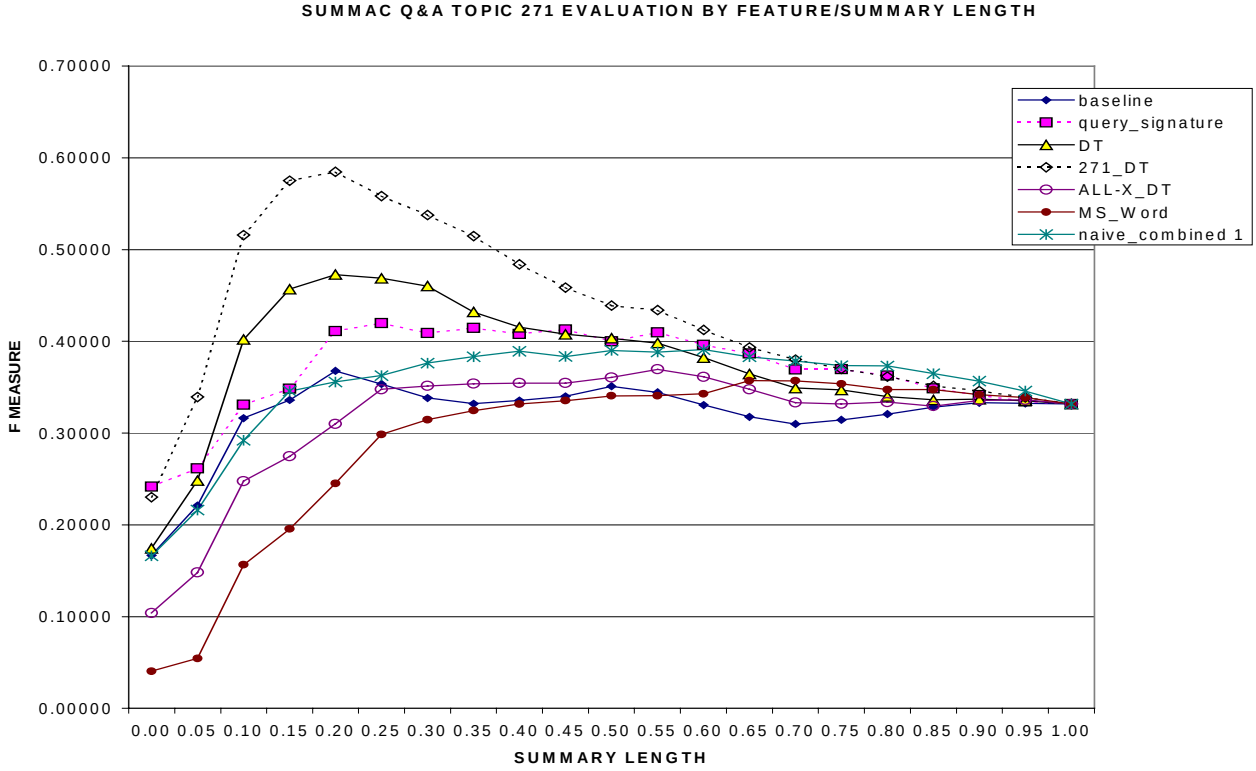

Figure 2(b). F-scores for selected parameters, Topic 271.

of separate system-internal parameters independently. By learning how each parameter contributes to the final result, one can approximate an glass-box study, which can be especially useful for the system builder; see for example [1].

We performed such a study with SUMMARIST, in a series of measurements. The training data derives from the Question and Answering summary evaluation data provided by TIPSTER-SUMMAC [42]. The TREC data is a collection of texts, classified into various topics, used for formal evaluations of Information Retrieval systems in a series of annual comparisons; see [12, 44]. This data set contains essential text fragments (phrases, clauses, and sentences) which must be included in summaries to answer some TREC topics. These fragments are each judged by a human judge.

As described in Section 2, SUMMARIST employs several independent modules to assign a score to each sentence, and then combines the scores to decide which sentences to extract from the input text. One can gauge the efficacy of each module by comparing, for different amounts of extraction, how many 'good' sentences the module selects by itself. We rate a sentence as good simply if it also occurs in the ideal human-made extract, and measure it using combined recall and precision (F-score). We used four topics from the TIPSTER-SUMMAC evaluation corpus Question and Answer test ([10], see Table 1). One of them contains 48 texts; the rest of them each containing 30 texts. Model extracts are created automatically from sentences containing answer keys.

Topic ID	Short Description	Number of Texts	Topic Prominence of Majorities
151	Overcrowded prisons	48	The topic is a Subsidiary theme
257	Cigarette consumption	30	The topic is a Subsidiary theme
258	Computer security	30	The main theme is on the topic
271	Solar Power	30	The topic is a subsidiary theme

Table 1 Topics used in glass-box query-oriented/generic informative extract evaluation.

Since we could find no systematic comparison of the parameters typically used in extract systems and also contained in SUMMARIST, we included them all. In addition, since we were working with question-oriented summaries, we also included several additional question-related features, such as the presence of adjectives, proper nouns, weekdays, and numbers. These features are easily computed for words and/or sentences during preprocessing. Finally, we tested various ways of combining parameters and features, and for reference we included Microsoft's AutoSummarize, available in the Word-97 word processing application.

SUMMARIST produced extracts of the same texts separately for each parameter, feature, combination function, and Word's summarizer, for a series of extracts ranging from 0% to 100% of the original text. Here we show two graphs, for the 30 texts about Topic 271 of the Question and Answer test, in Figures 2(a) and 2(b). Graphs for other topics are in the Appendix[2]. Figure 2(a) plots the F-measure of each parameter, feature, and combination function, which are:

baseline: the simplest baseline method of scoring each sentences by its position in the text; first sentence highest score, last sentence lowest. Scores normalized between 0 and 1. Labeled *baseline* in Figures 2(a) and 2(b).

title: normalized count of open-class words in sentence that also occur in title. Each term that also occurs in the title gets a positive score 1, otherwise 0. This assumes words in the title are important. Labeled *title* in Figure 2(a).

***tf* and *tf·idf* scores**: following [23] and information retrieval research [36], terms with higher term frequency and *tf·idf* values are more important. Labeled *tf* and *tf·idf* in Figure 2(a).

position score: given to sentences in fixed position in text, following the assumption that sentence position correlates with importance in genres with regular structure [9, 21]. In our domain of newspaper articles, SUMMARIST assigns higher scores for terms in the first 4 paragraphs. Labeled *OPP* in Figure 2(a).

query signature: normalized score given to sentences depending on number of query words they contain. Users often have a particular topic in mind when they request summaries. A list of open-class words extracted from the query forms the query signature. Labeled *query_signature* in Figures 2(a) and 2(b).

IR signature: score given to sentences depending on number and scores of *IR signature* words included. In an information retrieval environment, terms that occur more often in the top *n* retrieved documents are more important for the user query. We call the *m* most salient terms (ranked by *tf·idf*) the *IR signature*. This is a variation of topic signature [20]. Labeled *IR_signature* in Figure 2(a).

sentence length: score reflecting length of sentences (counting words), normalized by length of longest sentence. Labeled *sentence_length* in Figure 2(a).

average lexical connectivity: score is the number of terms shared with other sentences divided by the total number of sentences in the text. This assumes a sentence sharing more terms with other sentences is more important. Labeled *average_lexical_connectivity* in Figure 2(a).

numerical data: boolean value 1 given to sentences that include a numerical expression. Labeled *has_CD_tag* in Figure 2(a).

proper name: boolean value 1 given to sentences that include a proper noun. Labeled *has_NNP_tag* in Figure 2(a).

pronoun and adjective: boolean values 1 given to sentences that include a pronoun (reflecting coreference connectivity) and

[2] We include only selected parameter graphs for other topics. Readers who are interested in view the full graphs can visit the following web site: http://www.isi.edu/~cyl/cikm99.html.

adjective (counted independently). Labeled *has_PRP_tag* and *has_JJS_tag* respectively in Figure 2(a).

weekday and month: boolean values 1 given to sentences that include weekdays and months (counted independently). Respectively labeled *has_WDY_tag* and *has_MON_tag* in Figure 2(a).

quotation: boolean value 1 given to sentences containing a quote (reflecting opinions and evaluations, useful for queries asking for such). Labeled *has_QT* in Figure 2(a).

first sentences: score given to first sentence of each paragraph, reflecting text order, normalized between 0 and 1. When all first sentences used, remaining sentences are scored in text order. Labeled *is_1st_sent_par* in Figure 2(a).

decision tree combination function: the scores for all the above parameters and features were combined by the automated learning process using decision tree. Labeled *DT* in Figures 2(a) and 2(b).

Microsoft summarizer: score given to sentences extracted by the summarizer included in Word, for each summary length, normalized between 0 and 1. Labeled *MS_Word* in Figures 2(a) and 2(b).

simple combination function: the scores of the parameters title, tfidf, OPP, query signature, IR signature, numerical data, proper name, pronoun, adjective, weekday, and month were combined by simply adding their normalized scores, without further renormalization or weighting. Labeled *naïve_combined_1* in Figures 2(a) and 2(b).

Several conclusions can be drawn directly. As expected, the top-scoring result is the automatically trained decision tree combination function, trained on the whole corpus in a three-way cross-validation. However, the naïve combination seems to be a clear winner for three other topics; see Figures 3(a), 4(a), and 5(a). This is really counter-intuitive. The result indicates that learning a decision tree for text extraction is possible, but is not always the best learning algorithm to use. The fact that a naïve combination function outperforms the decision tree in 3 out of 4 cases implies that the heuristics used in the evaluation are independent of each other. If this is the case, then Bayesian classifiers may be a better choice for this task [1, 17].

The query signature achieves the second-best score, also not surprising, since (for the topic *Solar Power*[3]) the query was informative enough to yield sufficient information that was reflected in the text. For the other topics, unfortunately, the query signature module did not do so well. For topic 257 (numerical data about cigarette consumption), the single best performer is *has_CD_tag*, which indicates a number. This result indicates that query type is important.

The third best score (up to the 20% length) was achieved by three methods: the IR signature, baseline, and simple combination function. The relatively good performance of the baseline method, found also for the other topics, was somewhat unexpected, although on consideration that the genre is news articles, perhaps not too surprising. In fact, we find most of the lead sentences are included in the model summaries to provide background information and ensure coherence.

We also observe an interesting point that the prominence of a topic affects the performance of various heuristics. For example, most of the documents in topic 258 are on topic according to Table 1. Correspondingly, the baseline algorithm does very well for the 10%–40% range, even outperforming query signatures. We also see most heuristics performing well below the 20% length on this topic.

Summaries usually do not contain quoted sentences. Evaluation results show this is always a bad strategy (only topic 271 has a worse one).

Selecting the first sentence from each paragraph is also not a good strategy. For example, AP newswire stories usually contain single-sentence paragraphs, while *Wall Street Journal* (WSJ) and *Financial Times* (FT) paragraphs are longer. Thus selecting the first sentence from each paragraph from AP texts is equivalent to using the baseline method. The evaluation results of topics 257[4] (AP:9, FT:12, WSJ:9), 258 (AP:22, FT:5, WSJ:3), and 271 (AP:3, FT:21, WSJ:6) show that this heuristic is almost always the worst performer, especially for topic 271, where only 10% of texts are from AP. In contrast, it does fairly for topic 151, which however is tightly correlated with baseline performance.

The shapes of the curves in general indicate that to be most useful, summaries should not be longer than about 35% and not shorter than about 15%; no 5% summary achieved an F-score of over 0.25. They also indicate that each heuristic possesses different strength at different summary lengths.

To describe the effects of various combination functions more closely, and to estimate how well these methods do, we refer to Figure 2(b). In this figure we add two new combination functions:

variation 1: to determine best possible performance, the decision tree learning process from Section 3.3.8 was trained on the test data itself (topic 271) and plotted as *271_DT* in Figure 2(b).

variation 2: to determine a more normal case, the learning process was trained on the three other topics, excluding topic 271, and plotted as *ALL-X_DT* in Figure 2(b).

The results are very informative. As expected, the learned decision tree that was trained on the same (test) data performed best (line *271_DT*). Interestingly, its F-score was still below 0.6. We identify three factors to be considered. First, we do not know for this topic how well people agree when they construct extracts. It may well be that the inter-human agreement rate also provides an F-score of around 0.6. In that case, the method is doing as well as can be done. We need reliable studies of inter-human agreement on summary (extract) creation for future work. Second, there is quite a large gap between the scores of the decision tree trained on the test data and the second-best scorer, the decision tree trained on all the data, including the test data (line *DT*). There is an almost equal gap between that and the

[3] Query signature for topic Solar Power is: <solar, power, extent, fossil, parts, major, fuel, fuels, energy, alternative, source, worldwide, purpose, purposes, development, world, countries, country>.

[4] The number after each acronym is the number of texts from that source.

decision tree trained on all data except the test data (line *ALL-X_DT*). Since the other topics involved other query types, this proves the topic and query sensitivity of the summarization process. Still, for this topic, the query signature performed very well. Third, the extract process is inherently limited; the system is not able to pick out from individual sentences just those portions that are maximally informative and recombine them into a more condensed summary. Being able to do so will not increase the scores of the methods, but will enable them to reach their maximum level more quickly, at a shorter summary length.

The learning curves for topic 271 in Figure 5 suggest that more training material is helpful. Figure 4 shows a very interesting point that the improved performance of the combination function trained on topics 151, 258, and 271 (excluding texts of topic 257) than just on topic 257 alone demonstrates the need for a more targeted training corpus.

Some sentences are included in the model summaries to ensure textual coherence, rather than because of their own importance. We need a training corpus in which the effects of coherence and importance are separated.

4. CONCLUSIONS

We conduct an in-depth study of the effectiveness of several summarization heuristics through glass-box generic-query-oriented informative extract evaluation[5]. The major differences with previous summary evaluation work [1, 5, 16, 17, 24, 28] are: (1) we evaluated on many features in various extract lengths; (2) we focused on query-oriented extracts, though the results of topic 258 can be used to estimate generic extract performance; (3) we used news texts covering several topics; (4) we trained and tested on informative summary corpus.

In the experiment, comparing the graphs for the four query topics, it was strikingly clear that no single parameter or feature performs best overall. A feature such as *numerical data* gives excellent results for queries requiring numerical answers (*how many?* queries), while the feature *weekday* excels for *when?* queries. We conclude that no single parameter or feature suffices for query-based summaries. The complexities of types of queries require more detailed query analysis and the development of an extraction strategy for each type.

Furthermore, surprisingly, the simple combination function of parameters and features almost always provides performance comparable to the trained decision tree combination function, and sometimes exceeds it. We conclude that we need to investigate other learning methods, such as the Bayesian classifiers used by Kupiec et al. [17] and Aone et al. [1].

5. ACKNOWLEDGEMENTS

The author would like to thank Eduard Hovy, Ulf Hermjakob, and anonymous reviewers for helpful comments on this work..

6. REFERENCES

[1] Aone, C., M.E. Okurowski, and J. Gorlinsky. 1998. Trainable, Scalable Summarization System using Robust NLP and Machine Learning. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL), Montreal, Canada, pp. 62–66.

[2] Baldwin, B. and T. Morton. 1998. Coreference-Based Summarization. *In* T. Firmin Hand and B. Sundheim (eds) TIPSTER-SUMMAC Summarization Evaluation. Proceedings of the TIPSTER Text Phase III Workshop. Washington, DC.

[3] Barzilay, R. and M. Elhadad. 1998. Using lexical chains for text summarization. Proceedings of the ACL-97/EACL-97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, pp. 10–17.

[4] Baxendale, P.B. 1958. Machine-Made Index for Technical Literature—An Experiment. IBM Journal (October): 354–361.

[5] Brandow, R. K. Mitze, and L. Rau. 1995. Automatic Condensation of Electronic Publishing Publications by Sentence Selection. Information Processing and Management 31 (5): 675–685.

[6] Brill, E. 1992. A Corpus-Based Approach to Language Learning. Ph.D. dissertation, University of Pennsylvania.

[7] Buckley, C. and C. Cardie. 1997. SMART Summarization System. *In* T. Firmin Hand and B. Sundheim (eds) TIPSTER-SUMMAC Summarization Evaluation. Proceedings of the TIPSTER Text Phase III Workshop. Washington, DC.

[8] Cowie, J., E. Ludovik, and H. Molina-Salgado. 1998. MINDS—Multi-lingual Interactive Document Summarization. Proceedings of the TIPSTER Text Phase III Workshop. Washington, DC.

[9] Edmundson, H.P. 1969. New Methods in Automatic Extraction. Journal of the ACM 16 (2): 264–285.

[10] Firmin Hand, T. and B. Sundheim. 1998. TIPSTER-SUMMAC Summarization Evaluation. Proceedings of the TIPSTER Text Phase III Workshop. Washington, DC.

[11] Goldstein, J. and M. Borger. 1998. Underline Project: Scout: Automated Query-Relevant Document Summarization. Proceedings of the TIPSTER Text Phase III Workshop. Washington, DC.

[12] Harman, D. (editor). 1995. Proceedings of the Fourth Text Retrieval Conference (TREC-4). NIST, Gaithersburg, MD.

[13] Hovy, E.H. and L. Wanner. 1996. Managing Sentence Planning Requirements. Proceedings of the Workshop on Gaps and Bridges in NL Planning and Generation, at ECAI Conference. Budapest, Hungary, pp. 53–58.

[14] Hovy, E.H. and C-Y. Lin. 1999. Automated Text Summarization in SUMMARIST. *In* I. Mani and M. Maybury (eds), Advances in Automated Text Summarization. Cambridge: MIT Press, pp. 81–94.

[15] Hovy, E.H. and H. Liu. 1999. The Power of Indicator Phrases for Automated Text Summarization. Submitted.

[16] Jing, H., R. Barzilay, K. McKeown, and M. Elhadad. 1998. Summarization Evaluation Methods: Experiments and Results. *In* E.H. Hovy and D. Radev (eds), Proceedings of the AAAI Spring Symposium on Intelligent Text Summarization, pp. 60–68.

[5] We see the result on topic 258 as generic type and others as query-oriented. The evaluation is informative since we use question-and-answering corpus.

[17] Kupiec, J., J. Pedersen, and F. Chen. 1995. A Trainable Document Summarizer. Proceedings of the Eighteenth Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR), Seattle, WA, pp. 68–73.

[18] Langkilde, I. and K. Knight. 1998. Generation that Exploits Corpus Knowledge. *In* Proceedings of the COLING/ACL Conference, Montreal, Canada, pp. 704–709.

[19] Lin, C-Y. 1995. Topic Identification by Concept Generalization. Proceedings of the Thirty-third Conference of the Association of Computational Linguistics (ACL-95), Boston, MA, pp. 308–310.

[20] Lin, C-Y. 1997. Robust Automated Topic Identification. Ph.D. dissertation, University of Southern California.

[21] Lin, C-Y. and E.H. Hovy. 1997. Identifying Topics by Position. Proceedings of the Applied Natural Language Processing Conference (ANLP-97), Washington, DC, pp. 283–290.

[22] Lin, C-Y. 1999. Machine Translation for Information Access across the Language Barrier: the MuST System. Proceedings of the Machine Translation Summit VII, MT in the Great Translation Era, Singapore.

[23] Luhn, H.P. 1959. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development: 159–165.

[24] Mani, I. and E. Bloedorn. 1998. Machine Learning of Generic and User-Focused Summarization. Proceedings of AAAI-98, pp. 821–826.

[25] Mani, I., E. Bloedorn, and Barbara Gates. 1998. Using Cohesion and coherence models for text summarization. Proceedings of the AAAI-98 Spring Symposium on Intelligent Text Summarization, pp. 69–76.

[26] Mann, W.C. and S.A. Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. Text 8 (3): 243–281.

[27] Marcu, D. 1997. The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. Ph.D. dissertation, University of Toronto.

[28] Marcu, D. 1998. Improving Summarization through Rhetorical Parsing Tuning. Proceedings of the COLING-ACL Workshop on Very Large Corpora. Montreal, Canada.

[29] McKeown, K.R. and D.R. Radev. 1995. Generating Summaries of Multiple News Articles. Proceedings of the 18th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR), Seattle, WA, pp. 74–82.

[30] Miike, S., E. Itoh, K. Ono, and K. Sumita. 1994. A Full-Text Retrieval System with Dynamic Abstract Generation Function. Proceedings of the 17th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR-94), pp. 152–161.

[31] Miller, G., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University.

[32] Paice, C.D. 1990. Constructing Literature Abstracts by Computer: Techniques and Prospects. Information Processing and Management 26 (1): 171–186.

[33] Quinlan, J.R. 1992. C4.5: Programs for Machine Learning. San Mateo, California: Morgan Kaufmann Publishers.

[34] Rau, L.S. and P.S. Jacobs. 1991. Creating Segmented Databases from Free Text for Text Retrieval. Proceedings of the Fourteenth Annual ACM Conference on Research and Development in Information Retrieval (SIGIR), pp. 337–346. New York, NY.

[35] Reimer, U. and U. Hahn. 1998. A Formal Model of Text summarization Based on Condensation Operators of a Terminological Logic. *In* I. Mani and M. Maybury (eds), Advances in Automated Text Summarization. Cambridge: MIT Press.

[36] Salton, G. 1988. Automatic Text Processing. Reading, MA: Addison-Wesley.

[37] Salton, G., A. Singhal, M. Mitra, and C. Buckley. 1997. Automatic Text Structuring and Summarization. Information Processing and Management 33 (2): 193–208.

[38] Selman, B., H. Levesque, and D. Mitchell. 1992. A New Method for Solving Hard Satisfiability Problem. Proceeding of the Tenth National Conference on Artificial Intelligence (AAAI-92), pp. 440–446. San Jose, California.

[39] Spärck Jones, K. 1999. Introduction to Text Summarisation. *In* I. Mani and M. Maybury (eds), Advances in Automated Text Summarization. Cambridge: MIT Press.

[40] SPSS 1997. SPSS Base 7.5 Application Guide. SPSS Inc., Chicago.

[41] Strzalkowski, T. Jin Wang, and Bowden Wise. 1998. A Robust Practical Text Summarization. Proceedings of the AAAI-98 Spring Symposium on Intelligent Text Summarization, pp. 26–33.

[42] Sundheim, B. 1998. The TIPSTER Question-and-Answer (Q&A) Summarization Task: Test Design and Test

[43] Results. Proceedings of the TIPSTER Text Phase III Workshop. Washington, DC.

[44] Teufel, S. and M. Moens. 1998. Sentence Extraction as a Classification Task. *In* I. Mani and M. Maybury (eds), Advances in Automated Text Summarization. Cambridge: MIT Press.

[45] Voorhees, E. and D. Harman (editors). 1998. Proceedings of the Seventh Text Retrieval Conference (TREC-7). NIST, Gaithersburg, MD.

In *Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management* (CIKM), Kansas City, Kansas, Nov 2-6, 1999.

[46] Wnek, K., Bloedorn, E., and Michalski, R. 1995. Selective Inductive Learning Method AQ15C. George Mason University, Fairfax, Social Science Computing Review, 2 (Winter 1992), 453-469.

7. APPENDIX

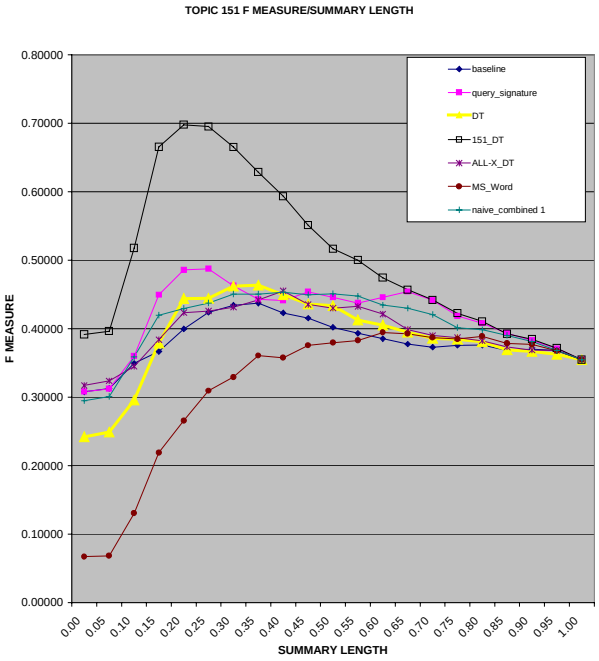

Figure 3. F-scores for selected parameters, Topic 151.

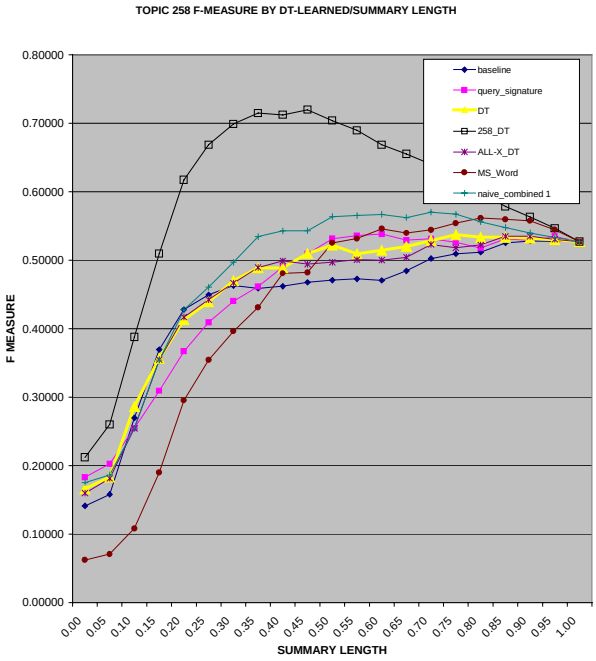

Figure 5. F-scores for selected parameters, Topic 258.

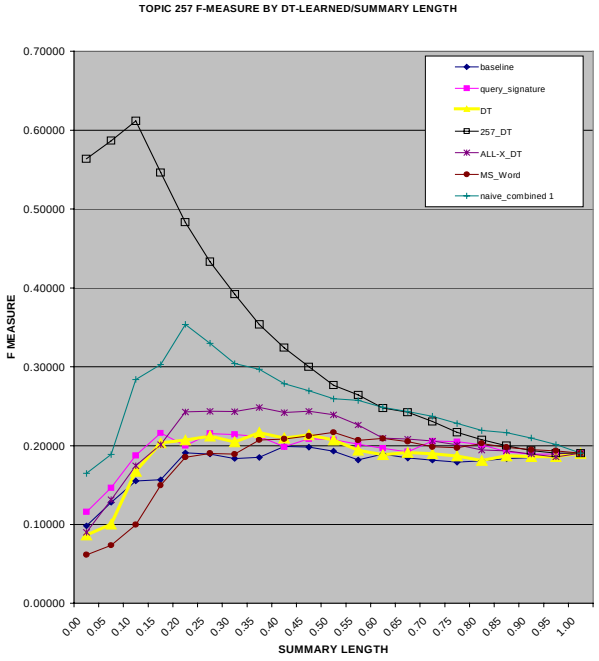

Figure 4. F-scores for selected parameters, Topic 257.

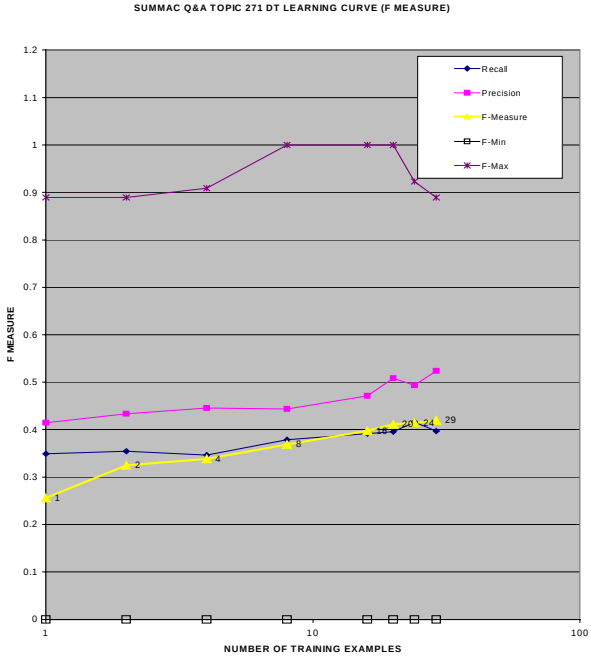

Figure 6. Learning curves for topic 271.