A Formal Model for Information Selection in Multi-Sentence Text Extraction

Elena Filatova
Department of Computer Science
Columbia University
New York, NY 10027, USA
filatova@cs.columbia.edu

Vasileios Hatzivassiloglou
Center for Computational Learning Systems
Columbia University
New York, NY 10027, USA
vh@cs.columbia.edu

Abstract

Selecting important information while accounting for repetitions is a hard task for both summarization and question answering. We propose a formal model that represents a collection of documents in a two-dimensional space of textual and conceptual units with an associated mapping between these two dimensions. This representation is then used to describe the task of selecting textual units for a summary or answer as a formal optimization task. We provide approximation algorithms and empirically validate the performance of the proposed model when used with two very different sets of features, words and atomic events.

1 Introduction

Many natural language processing tasks involve the collection and assembling of pieces of information from multiple sources, such as different documents or different parts of a document. Text summarization clearly entails selecting the most salient information (whether generically or for a specific task) and putting it together in a coherent summary. Question answering research has recently started examining the production of multi-sentence answers, where multiple pieces of information are included in the final output.

When the answer or summary consists of multiple separately extracted (or constructed) phrases, sentences, or paragraphs, additional factors influence the selection process. Obviously, each of the selected text snippets should individually be important. However, when many of the competing passages are included in the final output, the issue of information overlap between the parts of the output comes up, and a mechanism for addressing redundancy is needed. Current approaches in both summarization and long answer generation are primarily oriented towards making good decisions for each potential part of the output, rather than examining whether these parts overlap. Most current methods adopt a statistical framework, without full semantic analysis of the selected content passages; this makes the comparison of content across multiple selected text passages hard, and necessarily approximated by the textual similarity of those passages.

Thus, most current summarization or long-answer question-answering systems employ two levels of analysis: a content level, where every textual unit is scored according to the concepts or features it covers, and a textual level, when, before being added to the final output, the textual units deemed to be important are compared to each other and only those that are not too similar to other candidates are included in the final answer or summary. This comparison can be performed purely on the basis of text similarity, or on the basis of shared features that may be the same as the features used to select the candidate text units in the first place.

In this paper, we propose a formal model for integrating these two tasks, simultaneously performing the selection of important text passages and the minimization of information overlap between them. We formalize the problem by positing a textual unit space, from which all potential parts of the summary or answer are drawn, a conceptual unit space, which represents the distinct conceptual pieces of information that should be maximally included in the final output, and a mapping between conceptual and textual units. All three components of the model are application- and task-dependent, allowing for different applications to operate on text pieces of different granularity and aim to cover different conceptual features, as appropriate for the task at hand. We cast the problem of selecting the best textual units as an optimization problem over a general scoring function that measures the total coverage of conceptual units by any given set of textual units, and provide general algorithms for obtaining a solution.

By integrating redundancy checking into the selection of the textual units we provide a unified framework for addressing content overlap that does not require external measures of similarity between textual units. We also account for the partial overlap of information between textual units (e.g., a single shared clause), a situation which is common in nat-

ural language but not handled by current methods for reducing redundancy.

2 Formal Model for Information Selection and Packing

Our model for selecting and packing information across multiple text units relies on three components that are specified by each application. First, we assume that there is a finite set T of *textual units* $t_1, t_2, \ldots, t_n$, a subset of which will form the answer or summary. For most approaches to summarization and question answering, which follow the extraction paradigm, the textual units t_i will be obtained by segmenting the input text(s) at an application-specified granularity level, so each t_i would typically be a sentence or paragraph.

Second, we posit the existence of a finite set C of *conceptual units* $c_1, c_2, \ldots, c_m$. The conceptual units encode the information that should be present in the output, and they can be defined in different ways according to the task at hand and the priorities of each system. Obviously, defining the appropriate conceptual units is a core problem, akin to feature selection in machine learning: There is no exact definition of what an important concept is that would apply to all tasks. Current summarization systems often represent concepts indirectly via textual features that give high scores to the textual units that contain important information and should be used in the summary and low scores to those textual units which are not likely to contain information worth to be included in the final output. Thus, many summarization approaches use as conceptual units lexical features like *tf*idf* weighing of words in the input text(s), words used in the titles and section headings of the source documents (Luhn, 1959; H.P.Edmundson, 1968), or certain cue phrases like *significant*, *important* and *in conclusion* (Kupiec et al., 1995; Teufel and Moens, 1997). Conceptual units can also be defined out of more basic conceptual units, based on the co-occurrence of important concepts (Barzilay and Elhadad, 1997) or syntactic constraints between representations of concepts (Hatzivassiloglou et al., 2001). Conceptual units do not have to be directly observable as text snippets; they can represent abstract properties that particular text units may or may not satisfy, for example, status as a first sentence in a paragraph or generally position in the source text (Lin and Hovy, 1997). Some summarization systems assume that the importance of a sentence is derivable from a rhetorical representation of the source text (Marcu, 1997), while others leverage information from multiple texts to re-score the importance of conceptual units across all the sources (Hatzivassiloglou et al., 2001).

No matter how these important concepts are defined, different systems use text-observable features that either correspond to the concepts of interest (e.g., words and their frequencies) or point out those text units that potentially contain important concepts (e.g., position or discourse properties of the text unit in the source document). The former class of features can be directly converted to conceptual units in our representation, while the latter can be accounted for by postulating abstract conceptual units associated with a particular status (e.g., first sentence) for a particular textual unit. We assume that each conceptual unit has an associated *importance weight* w_i that indicates how important unit c_i is to the overall summary or answer.

2.1 A first model: Full correspondence

Having formally defined the sets T and C of textual and conceptual units, the part that remains in order to have the complete picture of the constraints given by the data and summarization approach is the mapping between textual units and conceptual units. This mapping, a function $f : T \times C \rightarrow [0, 1]$, tells us how well each conceptual unit is covered by a given textual unit. Presumably, different approaches will assign different coverage scores for even the same sentences and conceptual units, and the consistency and quality of these scores would be one way to determine the success of each competing approach.

We first examine the case where the function f is limited to zero or one values, i.e., each textual unit either contains/matches a given conceptual feature or not. This is the case with many simple features, such as words and sentence position. Then, we define the total information covered by any given subset S of T (a proposed summary or answer) as

$$I(S) = \sum_{i=1,\ldots,m} w_i \cdot \delta_i \qquad (1)$$

where w_i is the weight of the concept c_i and

$$\delta_i = \begin{cases} 1, \text{if } \exists j \in \{1, \ldots, m\} \text{ such that } f(t_j, c_i) = 1 \\ 0, \text{otherwise} \end{cases}$$

In other words, the information contained in a summary is the sum of the weights of the conceptual units covered by at least one of the textual units included in the summary.

2.2 Partial correspondence between textual and conceptual units

Depending on the nature of the conceptual units, the assumption of a 0-1 mapping between textual and conceptual units may or may not be practical or even

feasible. For many relatively simple representations of concepts, this restriction poses no difficulties: the concept is uniquely identified and can be recognized as present or absent in a text passage. However, it is possible that the concepts have some structure and can be decomposed to more elementary conceptual units, or that partial matches between concepts and text are natural. For example, if the conceptual units represent named entities (a common occurrence in list-type long answers), a partial match between a name found in a text and another name is possible; handling these two names as distinct concepts would be inaccurate. Similarly, an event can be represented as a concept with components corresponding to participants, time, location, and action, with only some of these components found in a particular piece of text.

Partial matches between textual and conceptual units introduce a new problem, however: if two textual units partially cover the same concept, it is not apparent to what extent the coverage overlaps. Thus, there are multiple ways to revise equation (1) in order to account for partial matches, depending on how conservative we are on the expected overlap. One such way is to assume minimum overlap (the most conservative assumption) and define the total information in the summary as

$$I(S) = \sum_{i=1,\ldots,m} w_i \cdot \max_j f(t_j, c_i) \qquad (2)$$

An alternative is to consider that $f(t_j, c_i)$ represents the extent of the $[0, 1]$ interval corresponding to concept c_i that t_j covers, and assume that the coverage is spread over that interval uniformly and independently across textual units. Then the combined coverage of two textual units t_j and t_k is

$$f(t_j, c_i) + f(t_k, c_i) - f(t_j, c_i) \cdot f(t_k, c_i)$$

This operator can be naturally extended to more than two textual units and plugged into equation (2) in the place of the max operator, resulting into an equation we will refer to as equation (3). Note that both of these equations reduce to our original formula for information content (equation (1)) if the mapping function f only produces 0 and 1 values.

2.3 Length and textual constraints

We have provided formulae that measure the information covered by a collection of textual units under different mapping constraints. Obviously, we want to maximize this information content. However, this can only sensibly happen when additional constraints on the number or length of the selected textual units are introduced; otherwise, the full set of available textual units would be a solution that proffers a maximal value for equations (1)–(3), i.e., $\forall S \subset T, I(S) \leq I(T)$. We achieve this by assigning a cost p_i to each textual unit t_i, $i = 1, \ldots, n$, and defining a function P over a set of textual units that provides the total penalty associated with selecting those textual units as the output. In our abstraction, replacing a textual unit with one or more textual units that provide the same content should only affect the penalty, and it makes sense to assign the same cost to a long sentence as to two sentences produced by splitting the original sentence. Also, a shorter sentence should be preferable to a longer sentence with the same information content. Hence, our operational definitions for p_i and P are

$$p_i = \text{length}(t_i), \quad P(S) = \sum_{t_i \in S} p_i$$

i.e., the total penalty is equal to the total length of the answer in some basic unit (e.g., words).

Note however, than in the general case the p_i's need not depend solely on the length, and the total penalty does not need to be a linear combination of them. The cost function can depend on features other then length, for example, number of pronouns—the more pronouns used in a textual unit, the higher the risk of dangling references and the higher the price should be. Finding the best cost function is an interesting research problem by itself.

With the introduction of the cost function $P(S)$ our model has two generally competing components. One approach is to set a limit on $P(S)$ and optimize $I(S)$ while keeping $P(S)$ under that limit. This approach is similar to that taken in evaluations that keep the length of the output summary within certain bounds, such as the recent major summarization evaluations in the Document Understanding Conferences from 2001 to the present (Harman and Voorhees, 2001). Another approach would be to combine the two components and assign a composite score to each summary, essentially mandating a specific tradeoff between recall and precision; for example, the total score can be defined as a linear combination of $I(S)$ and $P(S)$, in which case the weights specify the relative importance of coverage and precision/brevity, as well as accounting for scale differences between the two metrics. This approach is similar to the calculation of recall, precision, and F-measure adopted in the recent NIST evaluation of long answers for definitional questions (Voorhees, 2003). In this paper, we will follow the first tactic of maximizing $I(S)$ with a limit on $P(S)$ rather than attempting to solve the thorny issues of

weighing the two components appropriately.

3 Handling Redundancy in Summarization

Redundancy of information has been found useful in determining what text pieces should be included during summarization, on the basis that information that is repeated is likely to be central to the topic or event being discussed. Earlier work has also recognized that, while it is a good idea to select among the passages repeating information, it is also important to avoid repetition of the same information in the final output.

Two main approaches have been proposed for avoiding redundancy in the output. One approach relies on grouping together potential output text units on the basis of their similarity, and outputting only a representative from each group (Hatzivassiloglou et al., 2001). Sentences can be clustered in this manner according to word overlap, or by using additional content similarity features. This approach has been recently applied to the construction of paragraph-long answers (e.g., (Blair-Goldensohn et al., 2003; Yu and Hatzivassiloglou, 2003)).

An alternative approach, proposed for the synthesis of information during query-based passage retrieval is the maximum marginal relevance (MMR) method (Goldstein et al., 2000). This approach assigns to each potential new sentence in the output a similarity score with the sentences already included in the summary. Only those sentences that contain a substantial amount of *new information* can get into the summary. MMR bases this similarity score on word overlap and additional information about the time when each document was released, and thus can fail to identify repeated information when paraphrasing is used to convey the same meaning.

In contrast to these approaches, our model handles redundancy in the output at the same time it selects the output sentences. It is clear from equations (1)–(3) that each conceptual unit is counted only once whether it appears in one or multiple textual units. Thus, when we find the subset of textual units that maximizes overall information coverage with a constraint on the total number or length of textual units, the model will prefer the collection of textual units that have minimal overlap of covered conceptual units. Our approach offers three advantages versus both clustering and MMR: First, it integrates redundancy elimination into the selection process, requiring no additional features for defining a text-level similarity between selected textual units. Second, decisions are based on the same features that drive the summarization itself, not on additional surface properties of similarity. Finally, because all decisions are informed by the overlap of conceptual units, our approach accounts for partial overlap of information across textual units. To illustrate this last point, consider a case where three features A, B, and C should be covered in the output, and where three textual units are available, covering A and B, A and C, and B and C, respectively. Then our model will determine that selecting any two of the textual units is fully sufficient, while this may not be apparent on the basis of text similarity between the three text units; a clustering algorithm may form three singleton clusters, and MMR may determine that each textual unit is sufficiently different from each other, especially if A, B, and C are realized with nearly the same number of words.

4 Applying the Model

Having presented a formal metric for the information content (and optionally the cost) of any potential summary or answer, the task that remains is to optimize this metric and select the corresponding set of textual units for the final output. As stated in Section 2.3, one possible way to do this is to focus on the information content metric and introduce an additional constraint, limiting the total cost to a constant. An alternative is to optimize directly the composite function that combines cost and information content into a single number.

We examine the case of zero-one mappings between textual and conceptual units, where the total information content is specified by equation (1). The complexity of the problem depends on the cost function, and whether we optimize $I(S)$ while keeping $P(S)$ fixed or whether we optimize a combined function of both of those quantities. We will only consider the former case in the present paper. We start by examining an artificially simple case, where the cost assigned to each textual unit is 1, and the function P for combining costs is their sum. In this case, the total cost is equal to the number of textual units used in a summary.

This problem, as we have formalized it above, is identical to the *Maximum Set Coverage* problem studied in theoretical computer science: given C, a finite set of weighted elements, a collection T of subsets of C, and an integer k, find those k sets that maximize the total number of elements in the union of T's members (Hochbaum, 1997). In our case, the zero-one mapping allows us to view each textual unit as a subset of the conceptual units space, containing those conceptual units covered by the textual unit, and k is the total target cost. Unfortunately, *maximum set coverage* is NP-hard, as it is

reducible to the classic *set cover* problem (given a finite set and a collection of subsets of that set, find the smallest subset of that collection whose members' union is equal to the original set) (Hochbaum, 1997). It follows that more general formulations of the cost function that actually are more realistic for our problem (such as defining the total cost as the sum of the lengths of the selected textual units and allowing the textual units to have different lengths) will also result in an NP-hard problem, as we can reduce these versions to the special case of *maximum set coverage*.

Nevertheless, the correspondence with maximum set coverage provides a silver lining. Since the problem is known to be NP-hard, properties of simple greedy algorithms have been explored, and a straightforward local maximization method has been proved to give solutions within a known bound of the optimal solution. The greedy algorithm for maximum set coverage has as follows: Start with an empty solution S, and iteratively add to the S the set T_i that maximizes $I(S \cup T_i)$. It is provable that this algorithm is the best polynomial approximation algorithm for the problem (Hochbaum, 1997), and that it achieves a solution bounded as follows

$$I(\text{OPT}) \geq I(\text{GREEDY}) \geq \left[1 - \left(1 - \frac{1}{k}\right)^k\right] I(\text{OPT})$$

$$> \left(1 - \frac{1}{e}\right) I(\text{OPT}) \approx 0.6321 \times I(\text{OPT})$$

where $I(\text{OPT})$ is the information content of the optimal summary and $I(\text{GREEDY})$ is the information content of the summary produced by this greedy algorithm.

For the more realistic case where cost is specified as the total length of the summary, and where we try to optimize $I(S)$ with a limit on $P(S)$ (see Section 2.3), we propose two greedy algorithms inspired by the algorithm above. Both our algorithms operate by first calculating a ranking of the textual units in decreasing order. This ranking is for the first algorithm, which we call *adaptive greedy algorithm*, identical to the ranking provided by the basic greedy algorithm, i.e., each textual unit receives as score the increase in $I(S)$ that it generates when added to the output, in the order specified by the basic greedy algorithm. Our second greedy algorithm (dubbed *modified greedy algorithm* below) modifies this ranking by prioritizing the conceptual units with highest individual weight w_i; it ranks first the textual unit that has the highest contribution to $I(S)$ while covering this conceptual unit with the highest individual weight, and then iteratively proceeds with the textual unit that has the highest contribution to $I(S)$ while covering the next most important unaccounted for conceptual unit.

Given the rankings of textual units, we can then produce an output of a given length by adopting appropriate stopping criteria for when to stop adding textual units (in order according to their ranking) to the output. There is no clear rule for conforming to a specific length (for example, DUC 2001 allowed submitted summaries to go over "a reasonable percentage" of the target length, while DUC 2004 cuts summaries mid-sentence at exactly the target length). As the summary length in DUC is measured in words, in our experiments we extracted the specified number of words out of the top sentences (truncating the last sentence if necessary).

5 Experiments

To empirically establish the effectiveness of the presented model we ran experiments comparing evaluation scores on summaries obtained with a baseline algorithm that does not account for redundancy of information and with the two variants of greedy algorithms described in Section 4. We chose summarization as the evaluation task because "ideal" output (prepared by humans) and methods for scoring arbitrary system output were available for this task, but not for evaluating long answers to questions.

Data We chose as our input data the document sets used in the evaluation of multidocument summarization during the Document Understanding Conference (DUC), organized by NIST in 2001 (Harman and Voorhees, 2001). This collection contains 30 test document sets, each containing approximately 10 news stories on different events; document sets vary significantly in their internal coherence. For each document set 12 human-constructed summaries are provided, 3 for each of the target lengths of 50, 100, 200, and 400 words. We selected DUC 2001 because unlike later DUCs, ideal summaries are available for multiple lengths. We consider sentences as our textual units.

Features In our experiments we used two sets of features (i.e., conceptual units). First, we chose a fairly basic and widely used set of lexical features, namely the list of words present in each input text. We set the weight of each feature to its *tf*idf* value, taking *idf* values from `http://elib.cs.berkeley.edu/docfreq/`.

Our alternative set of conceptual units was the list of weighted *atomic events* extracted from the input texts. An atomic event is a triplet consisting of two named entities extracted from a sentence and a connector expressed by a verb or an event-related noun that appears in-between these two named entities.

The score of the atomic event depends on the frequency of the named entities pair for the input text and the frequency of the connector for that named entities pair. Filatova and Hatzivassiloglou (2003) define the procedure for extracting atomic events in detail, and show that these triplets capture the most important relations connecting the major constituent parts of events, such as location, dates and participants. Our hypothesis is that using these events as conceptual units would provide a reasonable basis for summarizing texts that are supposed to describe one or more events.

Evaluation Metric Given the difficulties in coming up with a universally accepted evaluation measure for summarization, and the fact that judgments by humans are time-consuming and labor-intensive, we adopted an automated process for comparing system-produced summaries to the ideal summaries written by humans. The ROUGE method (Lin and Hovy, 2003) is based on n-gram overlap between the system-produced and ideal summaries. As such, it is a recall-based measure, and it requires that the length of the summaries be controlled in order to allow for meaningful comparisons. Although ROUGE is only a proxy measure of summary quality, it offers the advantage that it can be readily applied to compare the performance of different systems on the same set of documents, assuming that ideal summaries are available for those documents.

Baseline Our baseline method does not consider the overlap in information content between selected textual units. Instead, we fix the score of each sentence as the sum of *tf*idf* values or atomic event scores. At every step we choose the remaining sentence with the largest score, until the stopping criterion for summary length is satisfied.

Results For every version of our baseline and approximation algorithms, and separately for the *tf*idf*-weighted words and event features, we get a sorted list of sentences extracted according to a particular algorithm. Then, for each DUC document set we create four summaries of each suggested length (50, 100, 200, and 400 words) by extracting accordingly the first 50, 100, 200, and 400 words from the top sentences.

To evaluate the performance of our summarizers we compare their outputs against the human models of the corresponding length provided by DUC, using the ROUGE-created scores for unigrams. Since scores are not comparable across different document sets, instead of average scores we report the number of document sets for which one algorithm outperforms another. We compare each of our approximation algorithms (adaptive and modified greedy) to the baseline.

Length	*Events*	*tf*idf*
50	$+3$	0
100	$+4$	-4
200	$+2$	-4
400	$+5$	0

Table 1: Adaptive greedy algorithm versus baseline.

Length	*Events*	*tf*idf*
50	0	$+$ 7
100	$+4$	$+$ 4
200	$+8$	$+$ 6
400	$+2$	$+14$

Table 2: Modified greedy algorithm versus baseline.

Table 1 shows the number of data sets for which the adaptive greedy algorithm outperforms our baseline. This implementation of our information packing model improves the ROUGE scores in most cases when events are used as features, while the opposite is true when *tf*idf* provides the conceptual units. This may be partly explained because of the nature of the *tf*idf*-weighted word features: it is possible that important words cannot be considered independently, and that the repetition of important words in later sentence does not necessarily mean that the sentence offers no new information. Thus words may not provide independent enough features for our approach to work.

Table 2 compares our modified greedy algorithm to the baseline. In that case, the model offers gains in performance when both events and words are used as features, and in fact the gains are most pronounced with the word features. For both algorithms, the gains are generally minimal for 50 word summaries and most pronounced for the longest, 400 word summaries. This validates our approach, as the information packing model has a limited opportunity to alter the set of selected sentences when those sentences are very few (often one or two for the shortest summaries).

It is worth noting that in direct comparisons between the adaptive and modified greedy algorithm we found the latter to outperform the former. We found also events to lead to better performance than *tf*idf*-weighted words with statistically significant differences. Events tend to be a particularly good representation for document sets with well-defined constituent parts (such as specific participants) that cluster around a narrow event. Events not only give us a higher absolute performance when compared

to just words but also lead to more pronounced improvement when our model is employed. A more detailed analysis of the above experiments together with the discussion of advantages and disadvantages of our evaluation schema can be found in (Filatova and Hatzivassiloglou, 2004).

6 Conclusion

In this paper we proposed a formal model for information selection and redundancy avoidance in summarization and question-answering. Within this two-dimensional model, summarization and question-answering entail mapping textual units onto conceptual units, and optimizing the selection of a subset of textual units that maximizes the information content of the covered conceptual units. The formalization of the process allows us to benefit from theoretical results, including suitable approximation algorithms. Experiments using DUC data showed that this approach does indeed lead to improvements due to better information packing over a straightforward content selection method.

7 Acknowledgements

We wish to thank Rocco Servedio and Mihalis Yannakakis for valuable discussions of theoretical foundations of the set cover problem. This work was supported by ARDA under Advanced Question Answering for Intelligence (AQUAINT) project MDA908-02-C-0008.

References

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization*, Spain.

Sasha Blair-Goldensohn, Kathleen R. McKeown, and Andrew Hazen Schlaikjer. 2003. Defscriber: A hybrid system for definitional qa. In *Proceedings of 26th Annual International ACM SIGIR Conference*, Toronoto, Canada, July.

Elena Filatova and Vasileios Hatzivassiloglou. 2003. Domain-independent detection, extraction, and labeling of atomic events. In *Proceedings of Recent Advances in Natural Language Processing Conference, RANLP*, Bulgaria.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based extractive summarization. In *Proceedings of ACL Workshop on Summarization*, Barcelona, Spain, July.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Jamie Callan. 2000. Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 165–172.

Donna Harman and Ellen Voorhees, editors. 2001. *Proceedings of the Document Understanding Conference (DUC)*. NIST, New Orleans, USA.

Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown. 2001. Simfinder: A flexible clustering tool for summarization. In *Proceedings of workshop on Automatic Summarization, NAACL*, Pittsburg, USA.

Dorit S. Hochbaum. 1997. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 94–143. PWS Publishing Company, Boston, MA.

H.P.Edmundson. 1968. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 23(1):264–285, April.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of 18th Annual International ACM SIGIR Conference*, pages 68–73, Seattle, USA.

Chin-Yew Lin and Eduard Hovy. 1997. Identifying topic by position. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, Washington, DC.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.

H.P. Luhn. 1959. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, April.

Daniel Marcu. 1997. From discourse structures to text summaries. In *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Spain.

Simone Teufel and Marc Moens. 1997. Sentence extraction as a classification task. In *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarizaion*, Spain.

Ellen M. Voorhees. 2003. Evaluating answers to definition questions. In *Proceedings of HLT-NAACL*, Edmonton, Canada, May.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sapporo, Japan, July.