code-name DUTCHMAN: A Text Summarization System

Erik Osheim
Swarthmore College
osheim@sccs.swarthmore.edu

Daniel Sproul
Swarthmore College
sproul@sccs.swarthmore.edu

Abstract

Text summarization is an interesting and challenging problem in natural language processing, and one that has numerous potential applications in the realm of data-mining, text searching, and information retrieval. We have implemented a summarization system appropriate for articles and technical texts.

The system, code-named DUTCHMAN, attempts to identify which sentences in the document are most appropriate for inclusion in a summary based on analysis using key noun-phrases. The system employs WordNet in order to extend the notion of key phrases to key concepts.

The system, in its current instantiation, only achieves mediocre results, but our work does suggest some promising avenues for future research in text summarization.

1 Introduction

The general problem of text summarization is very broad and difficult: Given some document of arbitrary length, can we produce a second document of roughly constant length (ie. a few sentences) that conveys the general meaning of the original? How can we process a text to determine what the text is about, and then reformulate that information to produce a viable summary? Certainly, humans are able to do this, but this is typically contingent on our ability to not only parse and read, but also *understand* the document in question. Thus, to fully address text summarization in general, we would need to first solve a large number of difficult and currently unresolved natural language processing and artificial intelligence problems.

One option would be to use a knowledge base to identify semantic facts and topics in a document; without fully solving things like the symbol grounding problem, we can still hope to *understand* the text's subject. Summarizers which take this approach are known as symbolic summarizers. However, there are some difficulties with taking a heavily symbolic approach. First, since it requires a large knowledge base in order to function, the results do not generalize across languages well. Second, symoblic approaches are especially vulnerable to the depth vs. robustness trade-off. Simply put, systems that are created to analyze a certain type of document can restrict themselves to that domain, allowing them to make more assumptions about the source texts and thus perform better, at the expense of generality (Hovy, 2000). Since symbolic summarizers have to make a lot of assumptions about the text's content, they tend to do especially well when they can specialize; however, this makes a general summarization tool difficult to implement symbolically. Theme and topic recognition (two common methods of symbolic summarization) are staggeringly complex in the most general cases (Mani, 1998).

Fortunately, in certain domains, documents tend to contain sentences which are self-summarizing. For example, journal and newspaper articles, and technical documents, tend to begin with, and, more generally, contain sentences which address the purpose and nature of the document as a whole. We cannot expect this sort of sentence to be found in certain other domains, for example fiction, where no part of the text can be expected to pertain to the text as a whole. Many text summarization systems (eg. (Barker, 1998), (Szpakowicz, 1996)) choose to adopt such a restricted domain, and thus are able to exploit the self-summarizing nature of such documents.

Within this restricted domain, we can reformulate the problem of text summarization as follows: How do we select the best sentences for inclusion in a summary, and what do we do with these sentences after we have selected them? We have based our work on the work of the Text Summarization Group at the University of Ottowa

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 63–68
Computer Science Department, Swarthmore College

Department of Computer Science (Barker, 1998). Their general method involves identifying key noun phrases within a document, and then applying various heuristics to weight sentences based on key phrase frequency, then just concatenating the sentences in order to produce a summary.

Our text summarization system, code-named DUTCHMAN, is structured similarly, but we have extended the key phrase analysis to a form of conceptual analysis based on WordNet, allowing us to increase the emphasis placed on certain key phrases which are representative of general concepts in which other key phrases in the document participate. For example, in a paper about engines, given that *engine*, *camshaft*, and *piston* are all key phrases, the salience of the word *engine* will be increased, because *camshaft* and *piston* are both parts of engines.

2 Related Work

Due to renewed intereset in text summarization, several conferences have recently addressed the problem. From these talks, it is obvious that researchers somewhat divided over the best methods of text summarization. While many researchers favor statistical approaches similar to the one pursued in DUTCHMAN, there are also symbolic summarizers, which place more weight on trying to find important topics through world-level concepts (Hovy, 2000). These systems try to identify an underlying topic (or topics) before ranking phrases and sentences on their score. (**?**) In this context, DUTCHMAN is a statistical summarizer which utilizes symbolic information (via WordNet) in an attempt to improve its statistically generated keywords.

Most other projects that use symbolic information do so before their statistical processing, or do the two forms of processing independently and then attempt to integrate the results ((Szpakowicz, 1996), (Mani, 1998)). However, there are many varieites of symbolic summarizers; its unclear what the best use of ontologies is, especially given the depth/robustness trade-off. Some examples of symbolic summarization methods from the TIPSTER conference are:

- use a graph of theme nodes linked via a custom thesaurus (CIR).
- use sentences determined to be about frequently mentioned individuals via co-reference resolution (Penn)
- use morphological analysis, name tagging, and co-reference resolution to weight sentences (SRA)

Ad hoc summaries (undirected summaries like the kind DUTCHMAN generates) only comprise some of the goal of summarization systems. Most systems also support interactive summarization (after being questioned, i.e. (Mani, 1998) and (Szpakowicz, 1996)), and topic-specific summarization (summarization with regard to specific set of interests, i.e. terrorist activity (Mani, 1998)). These systems serve different purposes, but most summarization methods can be used fairly effectively in any of the realms. (Mani, 1998)

3 DUTCHMAN Base System

As noted, our general approach is to identify key noun phrases within a document which indicate that the sentences in which they participate might be relevant summary sentences, ie. might contain content which is relevant to the overall meaning of the text.

Given an input text, our basic algorithm is as follows:

1. Split the document into sentences
2. Apply part-of-speech tagging, text-chunking, and noun-phrase identification
3. Identify key noun-phrases based on frequency
4. Select sentences based on key phrase frequencies
5. Concatenate sentences to generate the final summary

Lacking sophistacated sentence splitting tools, DUTCHMAN currently relies on a simple and imperfect script to handle sentence chunking.

POS tagging, text chunking, and NP identification are accomplished using the pre-trained fnTBL rule sets which are distributed with the fnTBL system (fnTBL is a freely-distributed rule-based machine-learning tool commonly employed in natural language processing tasks; TBL stands for Transformation-Based Learning) (Florian, 2001).

The remainder of the base system was implemented using Python. After sentence chunking and NP identification, we construct a database of frequencies for each noun-phrase in the document. In addtion to noun-phrases identified by fnTBL, we also add to the database individual nouns and noun-phrases connected by prepositional phrases; in this manner, given the string "King of Prussia Mall",rather than merely adding "King" and "Prussia Mall" to the database, we also add "Mall" and "King of Prussia Mall",which are, one might imagine, the truly salient key-phrases for the document in question.

We then identify 10 noun-phrases as "key phrases",ie. phrases whose sentence content is likely to pertain to the overall meaning of the document, and thus make good summary sentences. In the base system, the key-phrases are chosen based purely on which noun-phrases have the greatest frequencies. The scores for noun-phrases which are contained in larger noun-phrases (eg. "King" is contained in "King of Prussia") are discounted somewhat by

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 63–68
Computer Science Department, Swarthmore College

the scores of their containing phrases. We used 10 keyphrases because, given the length of our test documents, this tended to cover about 10-25implementation might include dynamic selection based on a fixed target percentage, but DUTCHMAN does not currently support this.

Each sentence is then scored based on the summed weights of the key phrases in the sentence. When a key phrase occurs more than once within the same sentence, an interesting issue arises. The most obvious approach would be to simply multiply by the key phrase's count in the sentence, but the problem with this is that we would like to in some manner reward diversity of key phrases for our summary sentences. Thus, a sentence which contains a single key phrase twice ought to, on average, fair poorer than a sentence which contains two distinct key phrases once each. We accomplish this by multiplying by the square-root of the count rather than just the count; thus, additional instances of a key phrase increase the score of a sentence, but always by an amount less than the prior instance. The resulting scoring equation is as follows:

$$\text{score}(S) = \sum_{w \in S} \text{weight}(w) \cdot \sqrt{\text{freq}(w)}$$

The final summary is then generated by selecting the three highest-scoring sentences in the text, and concatenating them in the order in which they occur in the text. We found that since our algorithm tended to pick longer sentences from the text, choosing the best three tended to produce summaries which had fairly varied content, but with brevity. While the longer sentences naturally tend to score higher, it is still a useful result, as longer sentences are often used to link various indepentently occuring ideas within a text.

4 Key-Concept Analysis

We refine our key phrase analysis by generalizing to a notion of key concepts. Within a given text, many of the key phrases will in some manner be related to a similar concept. For example, in an article about engines, both pistons and camshafts are parts of an engine, and thus can be said to participate in the concept *engine*.

In order to implement our key concept analysis, we employed WordNet. WordNet is an ontology; it contains information linking words in the English language. It stores many different types of relationships, such as hypernymy, holonymy, synonymy, and sense. Langauge processing systems which take advantage of WordNet have information about words and language that is fundamentally richer than those that do not (Miller, 1998).

After experimenting with WordNet, and creating a prototype summarization system without it, we found that the only two relations which seemed to provide us with useful information for summarization were hypernymy and holonymy (-hypen and -hholn). Hypernymy identifies hierarchical "is a" relationships (thus a query for "tabby" might return "tabby IS cat IS mammal IS animal IS organism IS entity"), whereas holonymy returns a variety of containment relationships, eg. "is a" or "part of" in a non-hierarchical fashion (thus a query for "tabby" might return "tabby IS A cat", whereas a query for "piston" might return "piston PART OF reciprocating engine"). In order to facilitate interfacing DUTCHMAN with WordNet, we implemented a WordNet subsystem which we termed FERNANDO.

Because our summarizer is not generative, there was no good way to take advantage of noun phrases which WordNet found to be related to the article, but which did not appear in the article. Therefore, we use the WordNet analysis to reconsider the weights in the noun-phrase frequency database, giving added weight to those noun-phrases which represent concepts in which other noun-phrases in the document participate. Thus, if the words "cat", "tabby", and "calico" all appear in a document, the score for "cat" would be increased because both "tabby" and "calico" are identified as being kinds of "cat". We then select the 10 most salient key phrases based on the adjusted weights.

This modified algorithm is reflected by adding an extra step in relation to the base-system algorithm:

3a Use WordNet to modify key noun-phrase values based on key-concept analysis

Our algorithm generates a tree of noun phrases for each keyphrase. It then weights each noun phrase in the tree based on the number of keyphrases in whose trees it participates, and how directly it is linked to them. We wish to favor relationships which are closer to the source words, thus given the example "tabby IS cat IS mammal IS ...", if both "cat" and "mammal" occur in the document, we wish to increase the score for "cat" more than the score for "mammal",because we are seeking to achieve the correct level of generalization which encompasses the salient noun-phrases in the document and yet still addresses the meaning of the document as a whole. An article might contain "tabby","calico", "siamese", etc., and thus is most likely about cats and not about mammals. However, an article which contains not only those words but also "labrador" and "squirrel" is more likely about mammals; here, despite the fact that the score for "cat" was increased more than the score for "mammal" by the various types of cat, all words which are kinds of mammal contribute to "mammal",so in most cases "mammal" will win out over "cat".

For each noun-phrase considered in WordNet analysis, we must now compute a score offset. In essence, we need a decaying distance function for the relevant internode distances, which we achieve with a decaying exponential.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 63–68
Computer Science Department, Swarthmore College

	Average Score		
Document	*Random*	*no WordNet*	*WordNet*
ADA	1.2	3.6	2.4
SAUDI	1.0	4.4	2.8
GW	2.6	3.6	2.8
ENGINE	1.2	1.8	1.6
PIRATE	1.6	4.0	2.8
Average	1.52	3.48	2.48

Table 1: Human-assigned summary scores

To determine the score offset for each noun-phrase N, we then sum over each considered noun-phrases n, for each adding its originally computed frequency weight times the decaying distance function:

$$\Delta\text{score}(N) = \sum_{n \in \text{noun-phrases}} \text{freq}(n) \cdot \alpha^{\text{distance}(N,n)}$$

where α, a constant, was empirically chosen to be 0.7, a value which helped acheive the aforementioned desired level of concept generalization.

5 Results

One of the inherent difficulties of the text summarization problem is that it is rather difficult to evaluate quantitatively. What makes a summary "good" varies from person to person and from document to document. Nonetheless, some attempt can be made to evaluate the quality of a summary.

We selected a set of five test documents: an article about the Americans with Disabilities Act (ADA), a text regarding a Gulf-War era Iraqi occupation of a Saudi Arabian town (SAUDI), a brief biography of George W. Bush (GW), an excerpt from a text about engines (ENGINE), and a brief article about pirates (PIRATE). For each, we generated summaries both with and without using FERNANDO. In addition, in order to establish a baseline for our system, we generated summaries based on purely random selection of sentences.

Our first evaluation scheme involved getting a group of human evaluators to score each summary on a scale from 1 (bad) to 5 (good). The results of this evaluation are displayed in Table 1. It is sadly apparent that FERNANDO seems more to detract than add to the quality of a summary, but nonetheless both are notably better than the results acheived by random selection.

Our second evaluation scheme involved using precision and recall metrics. For each document, human evaluators identified a list of what they felt were the ten most relevant key noun phrases, and then each summary was scored for precision and recall of this list. We calculated precision as the percentage of nouns in the summary which were in the key phrase list; in this manner, we prevent the possible favoring of gibberish sentences like "Engine engine engine." Recall was simply the percentage of words in the list which were contained in the summary. In the text summarization domain, a good recall score will indicate that a summary addressed the major content elements of the document, whereas a good precision score indicates that a summary is targeted and concise. Arguably, recall is a more important metric than precision in this domain, but both convey meaning regarding the quality of a summary. The baseline (random summary) results are displayed in Table 2 and the actual summary results are displayed in Table 3.

	Random	
Document	*Precision*	*Recall*
ADA	25%	30%
SAUDI	13%	10%
GW	36%	50%
ENGINE	0%	0%
PIRATE	33%	60%
Average	21%	30%

Table 2: Baseline Precision and Recall

	no WordNet		*with WordNet*	
Document	*Precision*	*Recall*	*Precision*	*Recall*
ADA	57%	90%	61%	90%
SAUDI	42%	100%	50%	100%
GW	31%	40%	23%	40%
ENGINE	63%	40%	48%	40%
PIRATE	45%	50%	43%	40%
Average	48%	64%	45%	62%

Table 3: Precision and Recall for test documents

6 Discussion

On the surface, it appears as if incorporating WordNet into our system has made it slightly worse rather than better, as we get the same recall but, on average, slightly worse precision. However, the engine and George W. Bush texts presented unique challenges to summarization, being that the engine article contained many lists of engine parts and very few summary-relevant sentences, and the Bush text was just very short, which meant that there were not enough key words present to make our WordNet analysis particularly meaningful. This suggests that perhaps the size of our keyword set needs to be allocated dynamically based on document length rather than constant. Also, in neither of the two problem cases did the sentences in the article have any real unifying themes, other than a very shallow description ("biography of George Bush", or "Mechanic's Textbook") which was not actually present in the text. Thus, our use of WordNet depends upon the assumption that the general

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 63–68
Computer Science Department, Swarthmore College

concepts relating the keyphrases actually be relevant to the summary.

On a more qualitative level, the no WordNet vs. WordNet summaries tended to be similar, and in the Saudi and ADA cases the WordNet ones provided more thorough detail, according to the human readers. Thus, despite the disappointing figures, analysis with WordNet did seem to yield some positive results.

7 Future Improvements

It would be interesting to test the human readers to see which documents they believed would be easier or harder to summarize, and compare those figures to our precision and recall figures for summarization with and without WordNet. Based on the articles and summaries that we have seen, we would guess that the articles which were found to be more easily summarizable by human readers would be the ones that the WordNet-aided summarization system would do best on.

DUTCHMAN lacked pronoun resolution, which severely hindered its performance. Since most sufficiently complicated ideas will span multiple sentences, and subsequent references of salient noun phrases are typically substituted for pronouns, pronoun resolution is key to derivative summarization (creating a summary directly out of excerpts from the text). Thus, a system with pronoun resolution could see a signifigant jump in its effectiveness. Additionally, DUTCHMAN lacked robust sentence splitting utility, and was thus often forced to deal with sentence fragments rather than whole sentences. Incorporating a more viable sentence splitter would no doubt increase DUTCHMAN's performance as well.

Another direction would be to use WordNet on all the noun phrases instead of just the statistically signifigant ones. It seems like concept-webs such as were used in the CRI summarizer might be an interesting way to augment our statistical data (Mani, 1998). As was remarked earlier, we did not find examples of summarizers that did symbolic analysis on a statistically selected subset, and this could explain FERNDANDO's confusing inability to help DUTCHMAN.

Future tests would probably have to define a narrower type of text to summarize; as we discovered, ontological assumptions about content which were valid for certain articles were invalid for others– in particular, it doesn't seem like biographies or instructional texts tend to yeild to the same techniques as explanatory articles, which are written with a more specific goal in mind. A larger testing set, with a narrower range of article types, and a broader base of human readers, with statistics on how well the humans believed they summarized the articles, and comparisons of the sets of human-identified keywords, would all aid in evaluating a summarizer.

8 Conclusion

Code-name DUTCHMAN is a fairly reasonable text summarization system, for which further fine-tuning would no doubt produce better results. Our addition of key-concept analysis using WordNet has proved helpful in some subjective cases, and further refinement of this technique, combined with other uses of WordNet, could facilitate the production of better summaries.

It is not clear whether methods that generate summaries out of excerpts can overcome all difficulties. Since the technique is limited by the quality of summarization-grade sentences in the document, it will never be perfect for certain types of documents. This is a problem that non-productive summarizers have regardless of whether they are statistical or symbolic. Many summarizations, such as the popular Cliff Notes series, are designed to do more than just abbreviate the text, but to paraphrase and explain; it would be desirable to have a summarizer that could do this. However, we do not have any belief that a system like ours could function in this way without radical modifications.

References

Radu Florian and Grace Ngai. 2001. *fnTBL*. Johns Hopkins University, Baltimore, MD

Eduard Hovy, Chin-Yew Lin, Daniel Marcu. 2000. *SUMMARIST: Automated Text Summarization*. http://www.isi.edu/natural-language/projects/SUMMARIST.html

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. *Summarizing Text Documents: Sentence Selection and Evaluation Metrics*. Carnegie-Mellon University and Just Research, Pittsburgh, PA

Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Leo Obrst, Therese Firmin, Michael Chrzanowski, Beth Sundheim. 1998. *The TIPSTER SUMMAC Text Summarization Evaluation*. The Mitre Corporation, McLean, VA

George A. Miller, Christiane Fellbaum, Randee Tengi, Susanne Wolff, Pamela Wakefield, and Helen Langone. 1998. *WordNet: a lexical database for the English language*. Princeton University, Princeton, NJ

Ken Barker, Ylias Chali, Terry Copeck, Stan Matwin, and Stan Szpakowicz. 1998. *The Design of a Configurable Text Summarization System*. University of Ottowa, Ottowa, CA

Stan Szpakowicz, Ken Barker, Terry Copeck, J. F. Delannoy, and Stan Matwin. 1996. *Preliminary Validation of a Text Summarization Algorithm*. University of Ottowa, Ottowa, CA

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 63–68
Computer Science Department, Swarthmore College

Appendix A: Sample Summaries

Here we include four summaries, both with and without using FERNANDO for two documents, SAUDI and ENGINE, chosen to be representative of "good" documents (SAUDI) and "bad" documents (ENGINE) for summarization by DUTCHMAN.

SAUDI - No FERNANDO:

A fierce battle for this deserted coastal town ended today when forces from Saudi Arabia and the emirate of Qatar, backed by American artillery and air strikes, evicted Iraqi troops and tanks, and freed two trapped U.S. reconnaissance teams. Marine commanders explained that Saudi forces had responsibility for the defense of the border area around Khafji, a town of 45,000 people on the Persian Gulf about six miles south of the Kuwait frontier. Marines provided artillery support and air strikes from Cobra gunships, but did not participate in the on-again, off-again ground battle, an occasionally tense confrontation involving close-quarters encounters between tanks and troops in the middle of town.

SAUDI - With FERNANDO:

A fierce battle for this deserted coastal town ended today when forces from Saudi Arabia and the emirate of Qatar, backed by American artillery and air strikes, evicted Iraqi troops and tanks, and freed two trapped U.S. reconnaissance teams. Marine commanders explained that Saudi forces had responsibility for the defense of the border area around Khafji, a town of 45,000 people on the Persian Gulf about six miles south of the Kuwait frontier. But Marine Lt. Col. Garrett, supervising Marine fire teams, supporting the Saudi counter-strikes, met with the U.S. officer serving as liaison with the Saudi and Qatari forces, who checked with Admire and called a meeting to see if the Marine reconnaissance teams could be extracted using a Saudi tank attack as cover.

ENGINE - No FERNANDO:

With the exhaust valve closed and the intake valve open, the piston moves down in the cylinder as the engine crankshaft turns. The operation of the four stroke cycle style of engine depends on the timing of its valves and their condition, on the piston rings, and on the cylinder walls. This is the standard number per cylinder in almost all four stroke cycle engines, with the exception of some aircraft engines and racing car engines which have four valves per cylinder.

ENGINE - With FERNANDO:

The operation of the four stroke cycle style of engine depends on the timing of its valves and their condition, on the piston rings, and on the cylinder walls. This is the standard number per cylinder in almost all four stroke cycle engines, with the exception of some aircraft engines and racing car engines which have four valves per cylinder. This causes a partial vacuum in the crankcase to prevent oil from being forced out of the engine past the piston rings, oil seals and gaskets.

Appendix B: DUTCHMAN's Abstract

Here we have used the DUTCHMAN system to generate an alternate abstract of the DUTCHMAN paper; the original abstract, references, and appendices were excluded from the source document; FERNANDO was used. All things considered, this summary is terrible.

DUTCHMAN - With FERNANDO:

First, since it requires a large knowledge base in order to function, the results do not generalize across languages well. Many text summarization systems choose to adopt such a restricted domain, and thus are able to exploit the self-summarizing nature of such documents. A larger testing set, with a narrower range of article types, and a broader base of human readers, with statistics on how well the humans believed they summarized the articles, and comparisons of the sets of human-identified keywords, would all aid in evaluating a summarizer.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 63–68
Computer Science Department, Swarthmore College