Summarization of Noisy Documents: A Pilot Study

Hongyan Jing
IBM T.J. Watson Research Center
Yorktown Heights, NY
hjing@us.ibm.com

Daniel Lopresti
19 Elm Street
Hopewell, NJ
dpl@dlopresti.com

Chilin Shih
150 McMane Avenue
Berkeley Heights, NJ
cls@prosodies.org

Abstract

We investigate the problem of summarizing text documents that contain errors as a result of optical character recognition. Each stage in the process is tested, the error effects analyzed, and possible solutions suggested. Our experimental results show that current approaches, which are developed to deal with clean text, suffer significant degradation even with slight increases in the noise level of a document. We conclude by proposing possible ways of improving the performance of noisy document summarization.

1 Introduction

Previous work in text summarization has focused predominately on clean, well-formatted documents, i.e., documents that contain relatively few spelling and grammatical errors, such as news articles or published technical material. In this paper, we present a pilot study of noisy document summarization, motivated primarily by the impact of various kinds of physical degradation that pages may endure before they are scanned and processed using optical character recognition (OCR) software.

As more and more documents are now scanned in by OCR, an understanding of the impact of OCR on summarization is crucial and timely. The Million Book Project is one of the projects that uses OCR technology for digitizing books. Pioneered by researchers at Carnegie Mellon University, it aims to digitize a million books by 2005, by scanning the books and indexing their full text with OCR technology (http://www.archive.org/texts/millionbooks.php).

Understandably, summarizing documents that contain many errors is an extremely difficult task. In our study, we focus on analyzing how the quality of summaries is affected by the level of noise in the input document, and how each stage in summarization is impacted by the noise. Based on our analysis, we suggest possible ways of improving the performance of automatic summarization systems for noisy documents. We hope to use what we have learned from this initial investigation to shed light on future directions.

What we ascertain from studying the problem of noisy document summarization can be useful in a number of other applications as well. Noisy documents constitute a significant percentage of documents we encounter in everyday life. The output from OCR and speech recognition (ASR) systems typically contain various degrees of errors, and even purely electronic media, such as email, are not error-free. To summarize such documents, we need to develop techniques to deal with noise, in addition to working on the core algorithms. Whether we can successfully handle noise will greatly influence the final quality of summaries of such documents.

Some researchers have studied problems relating to information extraction from noisy sources. To date, this work has focused predominately on errors that arise during speech recognition, and on problems somewhat different from summarization. For example, Gotoh and Renals propose a finite state modeling approach to extract sentence boundary information from text and audio sources, using both n-gram and pause duration information (Gotoh and Renals, 2000). They found that precision and recall of over 70% could be achieved by combining both kinds of features. Palmer and Ostendorf describe an approach for improving named entity extraction by explicitly modeling speech recognition errors through the use of statistics annotated with confidence scores (Palmer and Ostendorf, 2001). Hori and Furui summarize broadcast news speech by extracting words from automatic transcripts using a word significance measure, a confidence score, linguistic likelihood, and a word concatenation probability (Hori and Furui, 2001).

There has been much less work, however, in the case of noise induced by optical character recognition. Early papers by Taghva, et al. show that moderate error rates have little impact on the effectiveness of traditional information retrieval measures (Taghva et al., 1996a; Taghva et al., 1996b), but this conclusion does not seem to apply to the task of summarization. Miller, et al. study the performance of named entity extraction under a variety of scenarios involving both ASR and OCR output (Miller et al., 2000), although speech is their primary interest. They found that by training their system on both clean and noisy input material, performance degraded linearly as a function of word error rates. They also note in their paper: "To our knowledge, no other information extraction technology has been applied to OCR material" (pg. 322).

An intriguing alternative to text-based summarization is Chen and Bloomberg's approach to creating summaries without the need for optical character recognition (Chen and Bloomberg, 1998). Instead, they extract indicative summary sentences using purely image-based techniques and common document layout conventions. While this is effective when the final summary is to be viewed onscreen by the user, the issue of optical character recognition must ultimately be faced in most applications of interest (e.g., keyword-driven information retrieval).

For the work we present in this paper, we performed a small pilot study in which we selected a set of documents and created noisy versions of them. These were generated both by scanning real pages via OCR and by using a filter we have developed that injects various levels of noise into an original source document. The clean and noisy documents were then piped through a summarization system. We tested different modules that are often included in such systems, including sentence boundary detection, part-of-speech tagging, syntactic parsing, extraction, and editing of extracted sentences. The experimental results show that these modules suffer significant degradation as the noise level in the document increases. We discuss the errors made at each stage and how they affect the quality of final summaries.

In Section 2, we describe our experiment, including the data creation process and various tests we performed. In Section 3, we analyze the results of the experiment and correlate the quality of summaries with noise levels in the input document and the errors made at different stages of the summarization process. We then discuss some of the challenges in summarizing noisy documents and suggest possible methods for improving the performance of noisy document summarization. We conclude with future work.

2 The Experiment

2.1 Data creation

We selected a small set of four documents to study in our experiment. Three of four documents were from the TREC corpus and one was from a Telecommunications corpus we collected ourselves (Jing, 2001). All are professionally written news articles, each containing from 200 to 800 words (the shortest document was 9 sentences and the longest was 38 sentences).

For each document, we created 10 noisy versions. The first five corresponded to real pages that had been printed, possibly subjected to a degradation, scanned at 300 dpi using a UMAX Astra 1200S scanner, and then OCR'ed with Caere OmniPage Limited Edition. These included:

clean The page as printed.

fax A faxed version of the page.

dark An excessively dark (but legible) photocopy.

light An excessively light (but legible) photocopy.

skew The clean page skewed on the scanner glass.

Note that because the faxed and photocopied documents were processed by running them through automatic page feeders, these pages can also exhibit noticeable skew. The remaining five sample documents in each case were electronic copies of the original that had had synthetic noise (single-character deletions, insertions, and substitutions) randomly injected at predetermined rates: 5%, 10%, 15%, 20%, and 25%.

In general, we want to study both real and synthetic noise. The arguments in favor of the former are quite obvious. The arguments in favor of the latter is that it is easier to control synthetic noise effects, and often they have exactly the same impact on the overall process as real noise. Even though the errors may be artificial, the impact on later processes is probably the same. For example, changing ”nuclear” to ”nZclear” does not reflect a common OCR error. But it does have the same effect – changing a word in the dictionary to a word that is no longer recognized. If the impact is identical and it is easier to control, then it is beneficial to use synthetic noise in addition to real noise.

A summary was created for each document by human experts. For the three documents from the TREC corpus, the summaries were generated by taking a majority opinion. Each document was given to five people who were asked to select 20% of the original sentences as the summary. Sentences selected by three or more of the five human subjects were included in the summary of the document. For the document from the Telecommunications corpus, an abstract of the document was provided by a staff writer from the news service. These human-created summaries are useful in evaluating the quality of the automatic summaries.

2.2 Summarization stages

We are interested in testing how each stage of a summarization system is affected by noise, and how this in turn affects the quality of the summaries. Many summarization approaches exist, and it would be difficult to study the effects of noise on all of them. However, the following stages are common to many summarization systems:

- Step 1: Tokenization. The main task here is to break the text into sentences. Tokens in the input text are also identified.

- Step 2: Preprocessing. This typically involves part-of-speech tagging and syntactic parsing. This step is *optional*; some systems do not perform tagging and parsing at all. Topic segmentation is deployed by some summarization systems, but not many.

- Step 3: Extraction. This is the main step in summarization, in which the automatic summarizer selects key sentences (sometimes paragraphs or phrases) to include in the summary.

- Step 4: Editing. Some systems post-edit the extracted sentences to make them more coherent and concise.

For each stage, we selected one or two systems that perform the task and tested their performance on both clean and noisy documents.

- For tokenization, we tested two tokenizers: one is a rule-based system that decides sentence boundaries based on heuristic rules encoded in the program, and the other one is a trainable tokenizer that uses a decision tree approach for detecting sentence boundaries and has been trained on a large amount of data.

- For part-of-speech tagging and syntactic parsing, we tested the English Slot Grammar (ESG) parser (McCord, 1990). The outputs from both tokenizers were tested on ESG. The ESG parser requires as input divided sentences and returns a parse tree for each input sentence, including a part-of-speech tag for each word in the sentence. The reason we chose a full parser such as ESG rather than a part-of-speech tagger and a phrase chunking system is that the summary editing system in Step 4 uses the output from ESG. Although many sentence extraction systems do not use full syntactic information, it is not rare for summarization systems that do use parsing output to use a full parser, whether it is ESG or a statistical parser such as Collin's, since such summarization systems often perform operations that need deep understanding of the original text.

- For extraction, we used a program that relies on lexical cohesion, frequency, sentence positions, and cue phrases to identify key sentences (Jing, 2001). The length parameter of the summaries was set to 20% of the number of sentences in the original document. The output from the rule-based tokenizer was used in this step. This particular extraction system does not use tagging and parsing.

- In the last step, we tested a cut-and-paste system that edits extracted sentences by simulating the revision operations often performed by professional abstractors (Jing, 2001). The outputs from all the three previous steps were used by the cut-and-paste system.

All of the summaries produced in this experiment were generic, single-document summaries.

3 Results and Analysis

In this section, we present results at each stage of summarization, analyzing the errors made and their effects on the quality of summaries.

3.1 OCR performance

We begin by examining the overall performance of the OCR process. Using standard edit distance techniques (Esakov et al., 1994), we can compare the output of OCR to the ground-truth to classify and quantify the errors that have arisen. We then compute, on a per-character and per-word basis, a figure for average precision (percentage of characters or words recognized that are correct) and recall (percentage of characters or words in the input document that are correctly recognized). As indicated in Table 1, OCR performance varies widely depending on the type of degradation. Precision values are generally higher than recall because, in certain cases, the OCR system failed to produce output for a portion of the page in question. Since we are particularly interested in punctuation due to its importance in delimiting sentence boundaries, we tabulate a separate set of precision and recall values for such characters. Note that these are uniformly lower than the other values in the table. Recall, in particular, is a serious issue; many punctuation marks are missed in the OCR output.

Table 1: OCR performance relative to ground-truth (average precision and recall).

	Per-Character					
	All Symbols		Punctuation		Per-Word	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
OCR.clean	0.990	0.882	0.869	0.506	0.963	0.874
OCR.light	0.897	0.829	0.556	0.668	0.731	0.679
OCR.dark	0.934	0.739	0.607	0.539	0.776	0.608
OCR.fax	0.969	0.939	0.781	0.561	0.888	0.879
OCR.skew	0.991	0.879	0.961	0.496	0.963	0.869

3.2 Sentence boundary errors

Since most summarization systems rely on sentence extraction, it is important to identify sentence boundaries correctly. For *clean* text, the reported accuracy of sentence boundary detection is usually above 95% (Palmer and Hearst, 1997; Reyner and Ratnaparkhi, 1997; Riley, 1989). However, detecting sentence boundaries in noisy documents is a serious challenge since punctuation and capitalization, which are important features in sentence boundary detection, are unreliable in noisy documents. As we have just noted, punctuation errors arise frequently in the OCR output of degraded page images.

We tested two tokenizers: one is a rule-based system and the other is a decision tree system. The experimental results show that for the clean text, the two systems perform almost equally well. Manual checking of the results indicates that both tokenizers made very few errors. There should be 90 sentence boundaries in total. The decision tree tokenizer correctly identified 88 of the sentence boundaries and missed two (precision: 100%; recall: 98%). The rule-based tokenizer correctly identified 89 of the boundaries and missed one (precision: 100%; recall: 99%). Neither system made any false positive errors (i.e., they did not break sentences at non-sentence boundaries).

For the noisy documents, however, both tokenizers made significant numbers of errors. The types of errors they made, moreover, were quite different. While the rule-based system made many false negative errors, the decision tree system made many false positive errors. Therefore, the rule-based system identified far fewer sentence boundaries than the truth, while the decision tree system identified far more than the truth.

Table 2: Sentence boundary detection results: total number of sentences detected and average words per sentence for two tokenizers. Tokenizer 1 is decision tree based, and tokenizer 2 is rule based.

	Tokenizer 1		Tokenizer 2	
	Sent	Words/sent	Sent	Words/sent
Original	88	23	89	22
Snoise.05	95	20	70	27
Snoise.10	97	20	69	28
Snoise.15	105	19	65	30
Snoise.20	109	17	60	31
Snoise.25	121	15	51	35
OCR.clean	77	23	82	21
OCR.light	119	15	64	28
OCR.dark	70	21	46	33
OCR.fax	78	26	75	27
OCR.skew	77	23	82	21

Table 2 shows the number of sentences identified by each tokenizer for different versions of the documents. As we can see from the table, the noisier the documents, the more errors the tokenizers made. This relationship was demonstrated clearly by the results for the documents with synthetic noise. As the noise rate increases, the number of boundaries identified by the decision tree tokenizer gradually increases, and the number of boundaries identified by the rule-based tokenizer gradually decreases. Both numbers diverge from truth, but they err in opposite directions.

The two tokenizers behaved less consistently on the OCR'ed documents. For OCR.light, OCR.dark, and OCR.fax, the decision tree tokenizer produced more sentence boundaries than the rule-based tokenizer. But for OCR.clean and OCR.skew, the decision tree tokenizer produced fewer sentence boundaries. This may be related to the noise level in the document. OCR.clean and OCR.skew contain fewer errors than the other noisy versions (recall Table 1). This indicates that the decision tree tokenizer tends to identify fewer sentence boundaries than the rule-based tokenizer for clean text or documents with very low levels of noise, but more sentence boundaries when the documents have a relatively high level of noise.

Errors made at this stage are extremely detrimental, since they will propagate to all of the other modules in a summarization system. When a sentence boundary is incorrectly marked, the part-of-speech tagging and the syntactic parsing are likely to fail. Sentence extraction may become problematic; for example, one of the documents in our test set contains 24 sentences, but for one of its noisy versions (OCR.dark), the rule-based tokenizer missed most sentence boundaries and divided the document into only three sentences, making extraction at the sentence level difficult at best.

Since sentence boundary detection is important to summarization, the development of robust techniques that can handle noisy documents is worthwhile. We will return to this point in Section 4.

3.3 Parsing errors

Some summarization systems use a part-of-speech tagger or a syntactic parser in their preprocessing steps.

We computed the percentage of sentences that ESG failed to return a complete parse tree, and used that value as one way of measuring the performance of the parser on the noisy documents. If the parser cannot return a complete parse tree, then it definitely fails to analyze the sentence; but even when a complete parse tree is returned, the parse can be wrong. As we can see from Table 3, a significant percentage of noisy sentences were not parsed. Even for the documents with synthetic noise at a 5% rate, around 60% of the sentences cannot be handled by the parser. This indicates that a full parser such as ESG is very sensitive to noise.

Even when ESG produces a complete parse tree for a noisy sentence, the result is incorrect most of times. For instance, the sentence *“Internet sites found that almost 90 percent collected personal information from youngsters”* was transformed to *“uInternet sites fo6ndha alQmostK0 pecent coll / 9ed pe?”* after adding synthetic noise at a 25% rate. For this noisy sentence, the parser returned a complete parse tree that marked the word *“sites”* as the main verb of the sentence, and tagged all the other words in the sentence as nouns.[1] Although a complete parse tree is returned in this case, it is incorrect. This explains the phenomenon that the parser returned a higher percentage of complete parse trees for documents with synthetic noise at the 25% rate than for documents with lower levels of noise.

Table 3: Percentage of sentences with incomplete parse trees. Sentence boundaries were first detected using Tokenizer 1 and Tokenizer 2.

	Tokenizer 1	Tokenizer 2
Original	10%	5%
Snoise.05	59%	58%
Snoise.10	69%	71%
Snoise.15	66%	81%
Snoise.20	64%	66%
Snoise.25	58%	76%
OCR.clean	2%	3%
OCR.light	46%	53%
OCR.dark	37%	43%
OCR.fax	37%	30%
OCR.skew	5%	6%

The above results indicate that syntactic parsers are very vulnerable to noise in a document. Even low levels of noise lead to a significant drop in performance.

3.4 Extract quality versus noise level

In the next step, we studied how the sentence extraction module in a summarization system is affected by noise in the input document. The sentence extractor we used (Jing, 2001) relies on lexical links between words, word frequency, cue phrases, and sentence positions to identify key sentences. The performance of the system is affected by noise in multiple dimensions: lexical links are less reliable in a noisy condition; cue phrases are likely to be missed due to noisy spelling; and word frequency is less accurate due to different noisy occurrences of the same word.

Evaluation of noisy document summaries is an interesting problem. Both intrinsic evaluation and extrinsic evaluation need to deal with noise effect on the quality of final summaries. For intrinsic evaluation, it is debatable whether clean human summaries or noisy document summaries (or both) should be used for comparison. There are two issues related to ’noisy’ human summaries: one, whether such summaries are obtainable, and two, whether such summaries should be used in evaluation. We note that it is already difficult for a human to recover the information in the noisy documents when the synthetic noise rate reached 10%. Therefore, noisy human summaries will not be available for documents with relatively high level of noise. Secondly, even though the original documents are noisy, it is desirable for the final summaries to be fluent and clean. Therefore, if our ultimate goal is to produce a fluent and clean summary, it benefits to compare the automatic summaries with such summaries rather than noisy summaries.

We compared the noisy automatic summaries with the clean human summaries by using three measures: unigram overlap between the automatic summary and the human-created summary, bigram overlap, and the simple cosine. These results are shown in Table 4. The unigram overlap is computed as the number of unique words occurring both in the extract and the ideal summary for the document, divided by the total number of unique words in the extract. Bigram overlap is computed similarly, replacing words with bigrams. The simple cosine is computed as the cosine of two document vectors, the weight of each element in the vector being $1/\sqrt{N}$, where N is the total number of elements in the vector.

Not surprisingly, summaries of noisier documents generally have a lower overlap with human-created summaries. However, this can be caused by either the noise in the document or poor performance of the sentence extraction system. To separate these effects and measure the performance of sentence extraction alone, we also computed the unigram overlap, bigram overlap, and cosine between each noisy document and its corresponding original text. These numbers are included in Table 4 in parentheses; they are an indication of the average noise level in a document. For instance, the table shows that 97% of words that occurred in OCR.clean documents also appeared in the original text, while only 62% of words that occurred in OCR.light appeared in the original. This indicates that OCR.clean is less noisy than OCR.light.

3.5 Abstract generation for noisy documents

To generate more concise and coherent summaries, a summarization system may edit extracted sentences. To study how this step in summarization is affected by noise, we tested a cut-and-paste system that edits extracted sentences by simulating revision operations often used by human abstractors, including the operations of removing phrases from an extracted sentence, and combining a reduced sentence with other sentences (Jing, 2001). This

[1] One reason might be that the tagger is likely to tag unknown words as nouns, and all the noisy words are considered unknown words.

Table 4: Unigram overlap, bigram overlap, and simple cosine between extracts and human-created summaries (the numbers in parentheses are the corresponding values between the documents and the original text).

	Unigram	Bigram	Cosine
Original	0.85 (1.00)	0.75 (1.00)	0.51 (1.00)
Snoise.05	0.55 (0.61)	0.38 (0.50)	0.34 (0.65)
Snoise.10	0.41 (0.41)	0.22 (0.27)	0.25 (0.47)
Snoise.15	0.25 (0.26)	0.10 (0.13)	0.20 (0.31)
Snoise.20	0.17 (0.19)	0.04 (0.07)	0.14 (0.23)
Snoise.25	0.18 (0.14)	0.04 (0.04)	0.09 (0.16)
OCR.clean	0.86 (0.97)	0.78 (0.96)	0.50 (0.93)
OCR.light	0.62 (0.63)	0.47 (0.55)	0.36 (0.65)
OCR.dark	0.81 (0.70)	0.73 (0.65)	0.38 (0.66)
OCR.fax	0.77 (0.84)	0.67 (0.79)	0.48 (0.86)
OCR.skew	0.84 (0.97)	0.74 (0.96)	0.48 (0.93)

cut-and-paste stage relies on the results from sentence extraction in the previous step, the output from ESG, and a co-reference resolution system.

For the clean text, the cut-and-paste system performed sentence reduction on 59% of the sentences that were extracted in the sentence extraction step, and sentence combination on 17% of the extracted sentences. For the noisy text, however, the system applied very few revision operations to the extracted (noisy) sentences. Since the cut-and-paste system relies on the output from ESG and co-reference resolution, which failed on most of the noisy text, it is not surprising that it did not perform well under these circumstances.

Editing sentences requires a deeper understanding of the document and, as the last step in the summarization pipeline, relies on results from all of the previous steps. Hence, it is affected most severely by noise in the input document.

4 Challenges in Noisy Document Summarization

In the previous section, we have presented and analyzed errors at each stage of summarization when applied to noisy documents. The results show that the methods we tested at every step are fragile, susceptible to failures and errors even with slight increases in the noise level of a document. Clearly, much work needs to be done to achieve acceptable performance in noisy document summarization. We need to develop summarization algorithms that do not suffer significant degradation when used on noisy documents. We also need to develop robust natural language processing techniques. For example, it will be useful to develop a sentence boundary detection system that can identify sentence breaks in noisy documents more reliably. One way to achieve this might be to retrain an existing system on tokenized noisy documents so that it will learn features that are indicative of sentence breaks in noisy documents. However, this is only applicable if the noise level in the documents is low. For document with high level of noise, such approach will not be effective.

In the remainder of this section, we discuss several issues in noisy document summarization, identifying the problems and proposing possible solutions. We regard this as a first step towards a more comprehensive study on the topic of noisy document summarization.

4.1 Choosing an appropriate granularity

It is important to choose an appropriate unit level to represent the summaries. For clean text, sentence extraction is a feasible goal since we can reliably identify sentence boundaries. For documents with very low levels of noise, sentence extraction is still possible since we can probably improve our programs to handle such documents. However, for documents with relatively high noise rates, we believe it is better to forgo sentence extraction and instead favor extraction of keywords or noun phrases, or generation of headline-style summaries. In our experiment, when the synthetic noise rate reached 10% (which is representative of what can happen when real-world documents are degraded), it was already difficult for a human to recover the information intended to be conveyed from the noisy documents.

Keywords, noun phrases, or headline-style summaries are informative indications of the main topic of a document. For documents with high noise rates, extracting keywords or noun phrases is a more realistic and attainable goal than sentence extraction. Still, it may be desirable to correct the noise in the extracted keywords or phrases, either before or after summarization. There has been past work on correcting spelling mistakes and errors in OCR output; these techniques could be useful for this purpose.

4.2 Using other information sources

In addition to text, target documents contain other types of useful information that could be employed in creating summaries. As noted previously, Chen and Bloomberg's image-based summarization technique avoids many of the problems we have been discussing by exploiting document layout features. A possible approach to summarizing noisy documents, then, might be to use their method to create an image summary and then apply OCR afterwards to the resulting page. We note, though, that it seems unlikely this would lead to an improvement of the overall OCR results, a problem which may almost certainly must be faced at some point in the process.

4.3 Assessing error rates without ground-truth

The quality of summarization is directly tied to the level of noise in a document. In this context, it would be useful to develop methods for assessing document noise levels without having access to the ground-truth. Intuitively, OCR may create errors that cause the output text to deviate from "normal" text. Therefore, one way of evaluating OCR output, in the absence of the original ground-truth, is to compare its features against features obtained from a large corpus of correct text. Letter trigrams (Church and Gale, 1991) are commonly used to correct spelling and OCR errors (Angell et al., 1983; Kuckich, 1992; Zamora et al., 1981), and can be applied to evaluate OCR output.

We computed trigram tables (including symbols and punctuation marks) for 10 days of AP news articles and evaluated the documents used in our experiment. The trigrams were computed on letters and Good-Turing estimation is used for smoothing. The values in the table are average trigram scores for each document set. As expected, OCR errors create rare or previously unseen trigrams that lead to higher trigram scores in noisy documents. As indicated in Table 5, the ground-truth (original) documents have the lowest average trigram score. These scores provide a relative ranking that reflects the controlled noise levels (Snoise.05 through Snoise.25), as well as certain of the real OCR data (OCR.clean, OCR.dark, and OCR.light).

Table 5: Average trigram scores.

	Trigram score
Original	2.30
Snoise.05	2.75
Snoise.10	3.13
Snoise.15	3.50
Snoise.20	3.81
Snoise.25	4.14
OCR.clean	2.60
OCR.light	3.11
OCR.dark	2.98
OCR.fax	2.55
OCR.skew	2.40

Different texts have very different baseline trigram scores. The ranges of scores for clean and noisy text overlap. This is because some documents contain more instances of frequent words than others (such as *"the"*), which bring down the average scores. This issue makes it impractical to use trigram scores in isolation to judge OCR output.

It may be possible to identify some problems if we scan larger units and incorporate contextual information. For example, a window of three characters is too small to judge whether the symbol @ is used properly: *a@b* seems to be a potential OCR error, but is acceptable when it appears in an email address such as *lsa@bbb.com*. Increasing the unit size will create sparse data problems, however, which is already an issue for trigrams.

In the future, we plan to experiment with improved methods for identifying problematic regions in OCR text, including using language models and incorporating grammatical patterns. Many linguistic properties can be identified when letter sequences are encoded in broad classes. For example, long consonant strings are rare in English text, while long number strings are legal. These properties can be captured when characters are mapped into carefully selected classes such as symbols, numbers, upper- and lower-case letters, consonants, and vowels. Such mappings effectively reduce complexity, allowing us to sample longer strings to scan for abnormal patterns without running into severe sparse data problems.

Our intention is to establish a robust index that measures whether a given section of text is "summarizable." This problem is related to the general question of assessing OCR output without ground-truth, but we shift the scope of the problem to ask whether the text is summarizable, rather than how many errors it may contain.

We also note that documents often contain logical components that go beyond basic text. Pages may include photographs and figures, program code, lists, indices, etc. Tables, for example, can be detected, parsed, and reformulated so that it becomes possible to describe their overall structure and even allow users to query them (Hu et al., 2000). Developing appropriate ways of summarizing such material is another topic of interest.

5 Conclusions and Future Work

In this paper, we have discussed some of the challenges in summarizing noisy documents. In particular, we broke down the summarization process into four steps: sentence boundary detection, preprocessing (part-of-speech tagging and syntactic parsing), extraction, and editing. We tested each step on noisy documents and analyzed the errors that arose. We also studied how the quality of summarization is affected by the noise level and the errors made at each stage of processing.

To improve the performance of noisy document summarization, we suggest extracting keywords or phrases rather than full sentences, especially when summarizing documents with high levels of noise. We also propose using other sources of information, such as document layout cues, in combination with text when summarizing noisy documents. In certain cases, it will be important to be able to assess the noise level in a document; we have begun exploring this question as well. Our plans for the future include developing robust techniques to address the issues we have outlined in this paper.

Lastly, we regard presentation and user interaction as a crucial component in real-world summarization systems.

Given that noisy documents, and hence their summaries, may contain errors, it is important to find the best ways of displaying such information so that the user may proceed with confidence, knowing that the summary is truly representative of the document(s) in question.

References

R. Angell, G. Freund, and P. Willet. 1983. Automatic spelling correction using a trigram similarity measure. *Information Processing and Management*, 19(4):255–261.

F. R. Chen and D. S. Bloomberg. 1998. Summarization of imaged documents without OCR. *Computer Vision and Image Understanding*, 70(3):307–320.

K. Church and W. Gale. 1991. Probability scoring for spelling correction. *Statistics and Computing*, 1:93–103.

Jeffrey Esakov, Daniel P. Lopresti, and Jonathan S. Sandberg. 1994. Classification and distribution of optical character recognition errors. In *Proceedings of Document Recognition I (IS&T/SPIE Electronic Imaging)*, volume 2181, pages 204–216, San Jose, CA, February.

Y. Gotoh and S. Renals. 2000. Sentence boundary detection in broadcast speech transcripts. In *Proceedomgs of ISCA Tutorial and Research Workshop ASR-2000*, Paris, France.

C. Hori and S. Furui. 2001. Advances in automatic speech summarization. In *Proceedings of the 7th European Conference on Speech Communication and Technology*, pages 1771–1774, Aalborg, Denmark.

Jianying Hu, Ramanujan Kashi, Daniel Lopresti, and Gordon Wilfong. 2000. A system for understanding and reformulating tables. In *Proceedings of the Fourth IAPR International Workshop on Document Analysis Systems*, pages 361–372, Rio de Janeiro, Brazil, December.

Hongyan Jing. 2001. *Cut-and-paste Text Summarization*. Ph.D. thesis, Department of Computer Science, Columbia University, New York, NY.

K. Kuckich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439.

M. McCord, 1990. *English Slot Grammar*. IBM.

D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. 2000. Named entity extraction from noisy input: Speech and OCR. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 316–324, Seattle, WA.

D. Palmer and M. Hearst. 1997. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–267, June.

D. D. Palmer and M. Ostendorf. 2001. Improving information extraction by modeling errors in speech recognizer output. In J. Allan, editor, *Proceedings of the First International Conference on Human Language Technology Research*.

J. C. Reyner and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington D.C.

M. Riley. 1989. Some applications of tree-based modelling to speech and language. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 339–352, Cape Cod, MA.

Kazem Taghva, Julie Borsack, and Allen Condit. 1996a. Effects of OCR errors on ranking and feedback using the vector space model. *Information Processing and Management*, 32(3):317–327.

Kazem Taghva, Julie Borsack, and Allen Condit. 1996b. Evaluation of model-based retrieval effectiveness with OCR text. *ACM Transactions on Information Systems*, 14:64–93, January.

E. Zamora, J. Pollock, and A. Zamora. 1981. The use of trigram analysis for spelling error detection. *Information Processing and Management*, 17(6):305–316.