Automatic Analysis of Call-center Conversations

Gilad Mishne*
Informatics Institute, University of Amsterdam
Kruislaan 403, 1098SJ Amsterdam
The Netherlands
gilad@science.uva.nl

David Carmel, Ron Hoory,
Alexey Roytman, Aya Soffer
IBM Research Lab in Haifa
Haifa 31905, Israel
{carmel,hoory,roytman,ayas}@il.ibm.com

ABSTRACT

We describe a system for automating call-center analysis and monitoring. Our system integrates transcription of incoming calls with analysis of their content; for the analysis, we introduce a novel method of estimating the domain-specific importance of conversation fragments, based on divergence of corpus statistics. Combining this method with Information Retrieval approaches, we provide knowledge-mining tools both for the call-center agents and for administrators of the center.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Languages, Management

Keywords

Call centers, automatic speech recognition

1. INTRODUCTION

The role of call-centers is becoming increasingly central to corporates in recent years; this is caused by two main factors. First, the increase in the importance of individuals to companies, and the drive of the latter to acquire and keep customers for the long haul. Second, the rapid pace of advances in technology creates an ever-growing gap between users and automated systems, prompting them to require more technical assistance.

Call-centers are a general term for help desks, information lines and customer service centers; among the services typically provided by these centers are customer support, operator services, inbound and outbound telemarketing, and web-based services. While the name "call center" implies voice conversations, it includes any dialog-based support system in which a user receives a service by a professional; indeed, some call-centers provide additional means of conversing, e.g. online chat services.

Recent years have witnessed significant improvements in technologies related to call-centers. Methods for Automatic Speech Re-cognition (ASR) are constantly improving; while the accuracy of ASR systems depends on the scenario and environment, state-of-the-art systems achieve better than 90% accuracy in transcription of clean, high quality speech and 70%-80% in transcription of telephone calls. In parallel, research in the fields of Information Retrieval (IR) and Text Analysis has been progressing rapidly, largely spawned by the growth of the world wide web and the amount of available information on the internet. The advances in ASR technology, which result in cleaner text than that previously produced by transcription systems, enables the application of text analysis methods to text transcribed from call-center voice conversations, opening new opportunities in this field.

In this paper, we describe a system that uses text analytics and search algorithms to effectively address two issues call-center administrators are facing:

- **Assisting the call-center agent.** The online nature of the dialog requires the expert to provide a quick solution to the user, both to preserve customer satisfaction and to shorten the dialog and reduce the costs of the support system. For this, the user's problem should be identified as quickly as possible, and presented to the expert in a way that will assist her in providing a solution to it.

- **Agent monitoring.** Dialog systems require continuous monitoring of incoming calls for many management purposes; examples are gathering statistics about the type of issues addressed, and preventing abuse of the call-center resources for private conversations.

The rest of the paper is organized as follows. In the remaining part of this section we give a brief overview of related systems and technologies. In Section 2 we describe our approach for using text analysis to address the two tasks defined above. Section 3 follows with a detailed description of the components of our system, including the text transcription process and the text analysis. In Section 4 we illustrate the system's usage with examples; we conclude in Section 5.

*Work done while visiting IBM Research Haifa.

CIKM'05, October 31–November 5, 2005, Bremen, Germany.

1.1 Related Work

Most text-analysis research related to call-centers is focused on Call Routing [5, 12] – the task of forwarding callers to the right resource in the call center. Here, IR techniques are typically used to compare the customer's request to a set of known, classified requests. The relatively restricted domain of this classification task results in good performance of many of the approaches in this area. A few systems use more complex text analysis for additional call-center administration, to perform tasks similar to the ones we describe (but with different approaches). Busemann *et al.* use shallow NLP and machine learning to select solutions for problems presented in emails sent to a support center [2]; Tan *et al.* use similar techniques for analyzing the amount of resources needed to handle types of issues found in call-center records. eResponder [4] provides an integrated solution for automatic responses to user questions. It stores question and answer pairs that have previously been asked and answered. These pairs can be used to either provide an immediate response to user questions, or to assist customer service representatives in drafting new responses to similar questions or to yet unanswered questions. When a new question arrives, the system searches its databases for similar questions as well as for relevant answers to this question and finds the most relevant Q&A pair based on both these measures. A similar system is presented in [10].

2. IMPORTANCE ESTIMATION IN CALL-CENTER TRAFFIC

As presented earlier, the two challenges we are facing are (1) assisting the agent in addressing the issue raised by the caller, and (2) monitoring the usage of the call-center resources. We argue that these tasks share a common ground: both are related to the *relative importance* of parts of the conversation. This can be illustrated with two examples,[1] given in Figures 1 and 2, (both taken from our corpus of transcribed calls to the IBM customer support center, described in Section 4).

⟨Operator⟩	Thank you for calling IBM customer service this is John am I speaking with Kate
⟨Caller⟩	Yeah you are
⟨Operator⟩	Morning how can I help you
⟨Caller⟩	Oh I'm I am trying to connect remotely... it's um ah of course on the worse day possible as always... prompting me for a new password I put one in and it won't accept it... no matter what I put in this happens every time it prompts me what is the problem... every single time it doesn't you know... I'm following all the password rules it doesn't matter it rejects... every password I put in and I have to call you guys okay I'm done venting can you help me

Figure 1: Beginning of typical call

To effectively provide the agent with tools assisting her to answer the user's needs, the main issue raised by the user should be detected. In the call shown in Figure 1, we see a typical description of a problem by a user; unlike written problem reports (such as emails to support centers), oral descriptions tend to be long, un-focused and repetitive. However, upon closer examination of the fragments of the text (separated by ellipses) we note that some of them tend to be more significant than other – e.g., "prompting me for a new password I put one in and it won't accept it" seems more relevant than other fragments of the text. Consequently, the problem of identifying the main issue presented by the user can be viewed as identifying the important fragments in the beginning of the conversation.

⟨Caller⟩	Serial number four four six six six six... yes ma'am that's easy to remember
⟨Operator⟩	Wait a minute okay
⟨Caller⟩	I'd say I'd be able to um mark of the beast plus one
⟨Operator⟩	The mark of a beast plus one
⟨Caller⟩	Yes ma'am, you never heard of the mark of a beast
⟨Operator⟩	No
⟨Caller⟩	Oh sure they been tough with you... yeah after the battle of Armageddon
⟨Operator⟩	mhm
⟨Caller⟩	when uh tribulation period when...

Figure 2: Off-topic section in a call

In the second example – Figure 2 – we see a call which has gone "off-track", discussing topics which are irrelevant to the call-center. To detect such abuse of the call-center resources, it is necessary to identify parts of calls (or complete conversations) which are of low relevancy to the domain of the call-center. This too can be viewed as a problem of importance estimation of call fragments, where the importance is defined as the pertinence of the fragment in the context of the call-center.

A simple way of estimating the significance level of a fragment of a call is to estimate the significance level of each word in the fragment and combine these individual values. In most text analysis systems, the significance of words is inversely related to their frequency: the more common a word is, the less significance it has. Thus, stopwords – words with very high frequency – are usually ignored when indexing text for retrieval purposes. However, this rule of thumb does not hold for the type of importance estimation we are facing. For example, in our data – which consists of conversations dealing with software and hardware issues – "important" fragments tend to contain a high concentration of common words such as "error", "password", and so on. However, it is certainly not the case that *all* common words are important – stopwords such as "the" are insignificant within almost any domain.

Therefore, estimating the significance level of a word requires an evaluation of how characteristic the word is to a specific domain, compared to other domains. Rather than global significance, we are actually estimating domain-specific word significance (and hence, domain-specific fragment significance.

More formally, we propose the following method for domain-specific significance estimation.

Word-level Significance Estimation. First, we pre-compute importance values for words; this is done once, offline, by

[1]For clarity we present manually transcribed versions of the examples.

comparing the frequency of a word in the domain-specific corpus to its frequency in an open-domain corpus. The comparison can be done using any statistical test for distribution difference such as chi-squared or log likelihood measures [7].

At the end of this stage, we have a list of words with their corresponding importance levels for the specific domain. If we normalize all values between -1 and 1, words with values close to 1 will be very indicative of the domain, words with values close to -1 will be very indicative of being unrelated to the domain, and words with values around 0 will be neither related nor unrelated to the domain. In addition to the usage of single words as the basic unit of significance estimation, it is also possible to use word n-grams.

Fragment Significance Estimation. Once the word-level domain-specific significance levels are computed, we can assign fragment-level significance levels in a straightforward manner: given a text fragment, the total importance is a combination of the importance values of the words in the fragment (e.g., a sum, product, maximal value etc). A benefit of this approach is that since the calculation of the individual values (which requires corpus comparison) is done offline, the fragment-level estimation is very fast: it involves only lookups in the pre-built lexicon, which is typically small enough to be kept in memory. This is particularly beneficial for call-center data analysis which must be performed on the fly to be effective.

Related Approaches. Our method is somewhat related to the sentence-level importance calculation described by Schiffman for news articles [13]. While that approach also makes use of pre-constructed lexicons, it is aimed at open-domain text rather than domain-specific; as a result, the significant words detected are "important in a general sense, without respect to a particular document ... [having] a global importance". Additionally, while Schiffman uses the fact that news articles tend to include more important information in their first sentences, we rely only on statistical corpus divergence (since important sections can appear in any section of a call-center conversation).

An additional related concept is Kleinberg's notion of "burstiness" [9], which uses changes in frequency counts over time to detect locally-important words within a text.

3. SYSTEM DESCRIPTION

We now give an overview of our call-center monitoring system, and provide technical details about its modules.

The information flow in our system is represented in Figure 3. Audio from the call-center is delivered to the IBM Transcription Server prototype where it is transcribed to plain text (see Section 3.1). This text, along with call metadata such as time-stamps and channel identification (to distinguish between the caller and the operator), is sent in turn to an annotator implemented within the IBM UIMA framework [6]. The annotator analyzes the text and marks it with additional knowledge, regarding the local significance of each part of the conversation; additional details about this stage and about UIMA are given in Section 3.2. Lastly, the annotated text is analyzed by various applications, described in Section 3.3.

3.1 Transcribing Call-center Data

The transcription server, used for transcribing the call-center data, is an IBM research prototype; it was built on top of existing IBM core WebSphere technology, in particular WebSphere Voice Server [14] and its major components such as the speech recognition engine facilities. The transcription server is able to get an audio stream or a URI to a prerecorded audio file, and transcribe it. The transcription output is an XML file, which includes the transcript and some meta-data; for example, each word has time-stamps of its beginning and its end. These time-stamps enable the synchronization of the audio and the text data. The server uses the latest Large Vocabulary Continuous Speech Recognition (LVCSR) technology developed in IBM research [8], to transcribe continuous spontaneous 6KHz speech recorded by a call-center into text. The output words are selected from a large US English vocabulary, which has a good coverage of the spoken language. In general, an LVCSR system includes an engine, a vocabulary, a language model and an acoustic model. Those define the scope of transcription, the language domain in which it is expected to be, and the acoustic environment of the recordings. The full process can be characterized by several steps: the input speech signal is first analyzed into a sequence of acoustic feature vectors, and then statistical methods are used to determine the most probable word sequence. In the latter stage, the acoustic model is used for computing the probability of the observed sequence of vectors given a word sequence and the language model for computing the probability of a word sequence, independently of the speech input. The recognition process is carried out by maximizing the product of the two (Bayes law).

The transcription server can be configured to apply unsupervised adaptation techniques, in order to further adapt the recognition to the speaker or environment, and thus significantly improve transcription accuracy. The adaptation is carried out by an iterative process – recognition is first performed using the speaker independent models and standard acoustic features. The results of this first pass over the speech data are then used to adapt the models or to adapt the acoustic features used in the recognition process. Subsequent passes over the data use the adapted models or adapted acoustic features. The adaptation process is applied in off-line transcription of a recorded call, but some variants of it can also be used for online transcription. In call-center calls, comprising of a dialog between an operator (agent) and a caller, speaker adaptation is carried out separately for the speech sections of the operator and the caller. The operator can have pre-trained models that are applied when recognizing his speech, even without adaptation. Hence, in the current transcription server prototype, in order to enable the usage of adaptation and pre-trained models, the input should comprise two separate channels. If that is not the case, future versions of the transcription server will be able to apply automatic speaker segmentation, in order to segment the two speakers automatically. Furthermore, identifying the speakers using automatic speaker identification will allow the server to use previously trained models on identified speech sections. The transcription word error rate (WER) for typical call-center data is 30%-40% using speaker independent models and 20%-30% using unsupervised adaptation. Generally, these levels of accuracy, which might be practically hard to read by a human, are more than suffi-

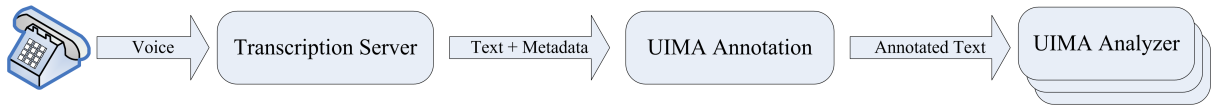

Figure 3: Information flow in the system

cient for textual access methods such as document indexing and retrieval or topic detection and tracking [1].

3.2 Data Annotation

Once the call is transcribed, we proceed to evaluating the importance of every fragment in the call as outlined in Section 2.

Comparing Corpora. For calculating the word-level significance values, we compare the frequency counts of the words in our corpus to a general corpus – the spoken component of the British National Corpus [11]). The comparison is done using a variation of the Mann-Whitney test. Table 1 shows the normalized values calculated for some example words in our data; words which are more common in our domain than in the general domain are assigned positive values, words which are more common in the general domain have negative values, and words which are equally common have zero or near-zero values.

click	0.997	the	0.000	born	-0.728
ibm	0.996	we	0.000	president	-0.772
password	0.995	he	0.000	council	-0.786
lotus	0.993	were	0.001	club	-0.812

Table 1: Sample word-level importance values

Call Segmentation. Ideally, the estimation of the significance should be carried out at the sentence level, or with grammatically-chunked parts of sentences; however, sentence delimiters are nonexistent in transcribed conversations, and grammatical analysis of it is very poor. A naive solution to dividing the call to segments is to choose a fixed-size window of words; this proves non-optimal for the domain, because of the large variation in sentence length which is typical of phone calls. We therefore take a heuristic approach to call segmentation, marking an "end of fragment" whenever the speaker has changed, or after a certain period of inactivity in the call. See for example Figure 1 where an ellipsis is used to delimit identified fragments.

Fragment-level significance. To assign a significance level to each fragment, we sum over the significance values of the words in the fragment, normalizing by the length of the fragment.

Both the call segmentation and the assignment of the significance level is done using the UIMA SDK.

UIMA. UIMA (Unstructured Information Management Architecture) is a software architecture for supporting applications that integrate search and analytics over a combination of structured and unstructured information; it has recently been publicly released. [2] UIMA is based on document-level analysis performed by component processing elements named Text Analysis Engines (TAEs). Examples of TAEs include language translators, document summarizers, named-entity detectors, and relationship extractors. Each TAE specializes in discovering specific concepts (or "semantic entities") implicit in the document text. The document analysis is based on a data structure, named the Common Analysis Structure (CAS), which contains the original document (the subject of analysis) and associated metadata in the form of annotation with respect the original text. The CAS is passed through an application-specified sequence of TAEs. Each TAE in the sequence considers the input CAS, potentially infers and adds additional annotations, and outputs an updated CAS. Annotations in the CAS are maintained separately from the document itself, and they often over-lap. A main goal of UIMA is to support "semantic search" – the capability to find documents based on the semantic content discovered by the TAEs. To this end, UIMA specifies search engine indexing and query interfaces: the indexing interface supports the indexing of tokens as well as the indexing of annotations; correspondingly, the query interface supports queries that may be predicated on nested or over-lapping structures of annotations and tokens in addition to Boolean combinations of tokens and annotations.

Using UIMA, we annotate each word in the text with its significance value, mark boundaries of fragments in the call, and annotate each fragment with its aggregated importance level. UIMA search is used to provide a basic search functionality over the transcribed data.

3.3 Call Analysis

At this stage the call is transcribed and annotated; the next stage in our system is an analysis of it for addressing the two issues we described, namely, assisting the agent and monitoring the call-center resources.

3.3.1 Suggesting Potential Solutions to Presented Problems

Many issues with which users address a call-center have been encountered previously. Typically, the call-center maintains a knowledge base of issues and their solutions; this gives the agents the ability to search or browse the collection of encountered issues, assisting them in addressing requests of types they are not familiar with.

As an extension to this helper application, we automatically detect the main issue addressed in the call, and retrieve possible solutions for it as the call progresses. We used the Juru IR engine [3] to index a collection of 7152 software and hardware issues and their solutions, manually developed by domain experts. An example issue from this collection is given in Figure 4. Each issue from the collection is indexed as a single document containing two fields: title (the problem) and body (the solution).

As the call topic, we simply take the first few fragments of

[2] http://www.alphaworks.ibm.com/tech/uima

Issue: Lotus 123 97 - Printing Cell Formulas Using Preview and Page Setup
Solution: From the menu bar, select File. Select Preview and Page Setup from the 123 menu options. Click on the Include tab in the Properties for Preview and Page Setup dialog box. Under Show, select Formulas. Note: When the workbook is printed, formulas and their locations will print on a separate page.

Figure 4: Sample problem/solution pair from our collection

the call that are "significant enough" according to our significance estimation model; for this, we define two thresholds: the minimal significance value needed for a fragment to be considered as part of the topic, and a maximum total significance value for the topic. During the incoming call, every fragment is analyzed for significance; if it exceeds the minimal value, it is added to the detected topic of the call, until the maximal threshold is reached. Typically, the topics span 1-3 fragments.

Once the topic is identified, we use it as a query to the Juru engine, searching both in the title and the body fields (and assigning a higher weight to the title). We display the top retrieved results in a pop-up window to the agent as she is listening to the call, enabling her to quickly scan possible solutions or different reformulations of the issue while the customer is describing the problem. An example of this process is shown in Section 4. In our experience, the queries tend to be relatively long, producing high-quality, relevant results.

3.3.2 *Detecting Off-topic Segments*

Given the analysis of the significance of each fragment in the call, identifying calls which are completely off-topic, or off-topic sections within "legitimate" calls, is straightforward. While variation in the importance level of sentences in a conversation (as well as occasional low-importance fragments) is normal, long segments within a call which have low significance are unusual, and typically indicate irrelevant calls or parts thereof. Therefore, we simply keep track of the significance of each fragment in the call, and detect continuous spans of fragments that fail to reach a minimal significance value. To illustrate this, Figure 5 shows two graphs of significance levels of fragments within calls: a standard call, and a call containing a long off-topic section.

4. EXAMPLES

Our corpus consists of 2276 calls made to the IBM internal customer support service during 2001; the calls deal with a range of software and hardware problems. The total number of words in the collection is 1.7M, and the unique number of words is 18385. Preliminary experimentation with the system on manually transcribed data shows promising results.

We show two typical screen-captures from our system: the display to the agent during the call, when the system identified the topic and retrieved possible solutions for it (Figure 6) and a highlighted off-topic segment, shown to the administrator (Figure 7).

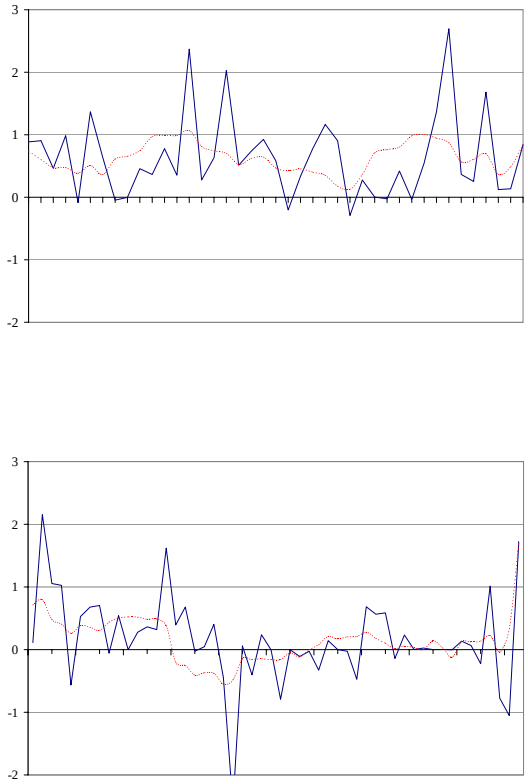

Figure 5: Significance levels of fragments throughout calls: typical dialog (top) and call containing off-topic segment (bottom). Continuous (blue) line is actual fragment significance; dashed (red) line marks local average.

5. CONCLUSIONS

We presented a system for monitoring call-center conversations, which uses text analytics to assist both the call-center agents and administrators. Our system provides an end-to-end solution, incorporating technologies for Automatic Speech Recognition, Text Analysis, and Information Retrieval. The main novelty of our system is a method for identifying the domain-specific importance levels of fragments in the call, and usage of this method both for retrieving possible solutions to the problem presented in the conversation, and for detecting abuse of the call-center resources.

To demonstrate our system in this work, we used manually-transcribed data; testing the effect of recognition errors on the system's performance is beyond the scope of this paper. Having said that, we anticipate that our algorithm will be fairly robust to moderate error rates, since it makes use of word redundancy within a fragment. While error rates of 20% may result in an incorrect word or two within a fragment (on average), this is unlikely to drastically change the significance level of the entire fragment. Similar robustness assumptions have been made in the IR domain on ASR-transcribed data, and were found correct [1].

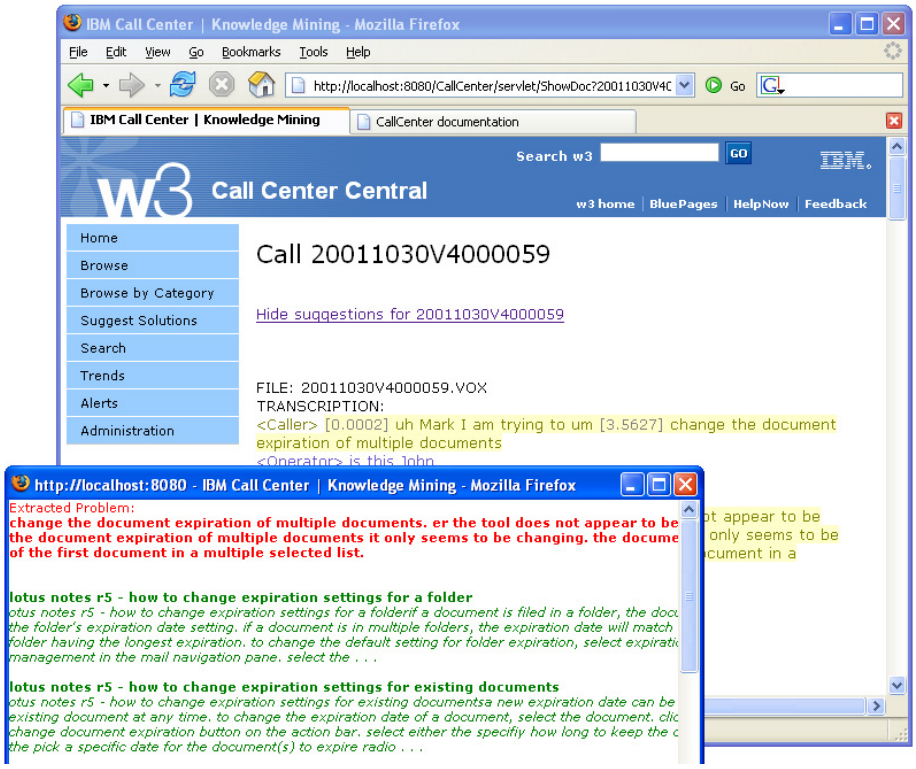

Figure 6: Retrieved solutions for a call

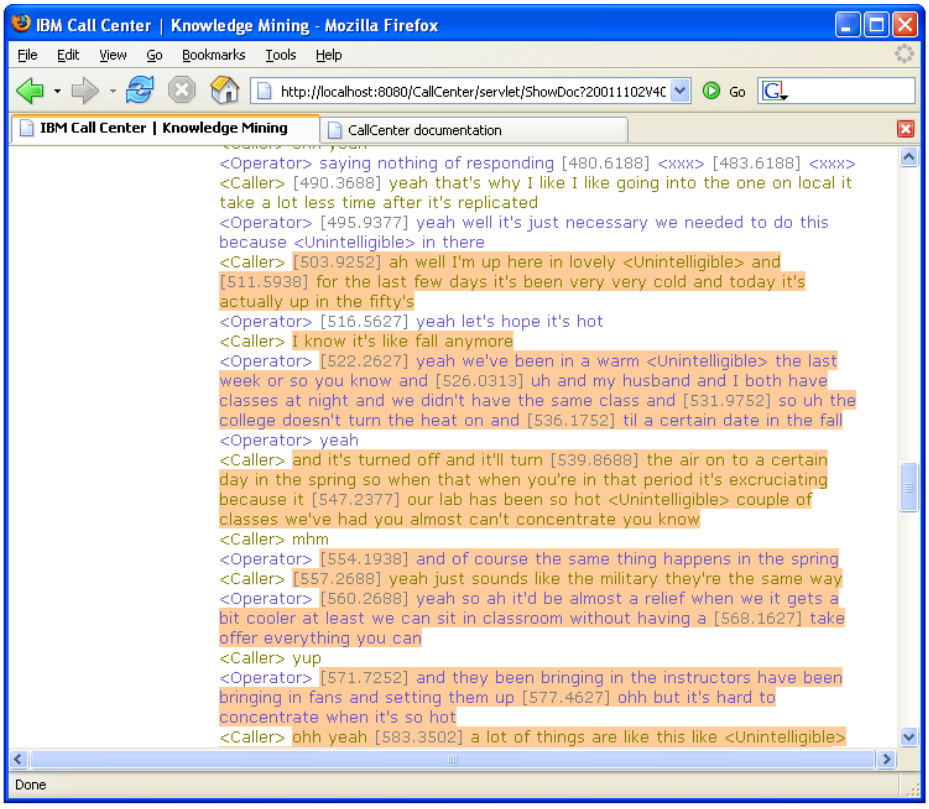

Figure 7: Highlighted possible off-topic segment

In the future, we plan a more rigorous evaluation of our system, including testing our algorithm on ASR-transcribed data and examining the effect of the recognition errors on its performance.

Acknowledgments. We thank Olivier Siohan from the IBM T.J. Watson research center for providing the required data and for assistance on ASR topics.

6. REFERENCES

[1] J. Allan. Perspectives on information retrieval and speech. In *Information Retrieval Techniques for Speech Applications*, pages 1–10. Springer, 2002.

[2] S. Busemann, S. Schmeier, and R. G. Arens. Message classification in the call center. In *Proceedings of the sixth conference on Applied natural language processing*, pages 158–165, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[3] D. Carmel, E. Amitay, M. Herscovici, Y. S. Maarek, Y. Petruschka, and A. Soffer. Juru at trec 10 - experiments with index pruning. In *TREC*, 2001.

[4] D. Carmel, M. Shtalhaim, and A. Soffer. eResponder: Electronic Question Responder. In *CooplS '00: Proceedings of the 7th International Conference on Cooperative Information Systems*, pages 150–161, London, UK, 2000. Springer-Verlag.

[5] J. Chu-Carroll and B. Carpenter. Vector-based natural language call routing. *Comput. Linguist.*, 25(3):361–388, 1999.

[6] D. Ferrucci and A. Lally. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3):476–489, 2004.

[7] A. Kilgarriff. Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):1–37, 2001.

[8] B. Kingsbury, L. Mangu, G. Saon, G. Zweig, S. Axelrod, V. Goel, K. Visweswariah, and M. Picheny. Towards domain-inependent conversational speech recognition. In *Eurospeech*, Geneva, Switzerland, September 2003.

[9] J. Kleinberg. Bursty and hierarchical structure in streams. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 91–101, New York, NY, USA, 2002. ACM Press.

[10] L. Kosseim, S. Beauregard, and G. Lapalme. Using information extraction and natural language generation to answer e-mail. *Data & Knowledge Engineering*, 38(1):85–100, 2001.

[11] G. Leech, P. Rayson, and A. Wilson. *Word Frequencies in Written and Spoken English: based on the British National Corpus*. Longman, 2001.

[12] G. Riccardi, A. Gorin, A. Ljolje, and M. Riley. A spoken language system for automated call routing. In *Proc. ICASSP '97*, pages 1143–1146, Munich, Germany, 1997.

[13] B. Schiffman. Building a Resource for Evaluating the Importance of Sentences. In *LREC02*, Las Palmas, Spain, May–June 2002.

[14] IBM WebSphere Voice Server. *http : //www.ibm.com/software/pervasive/voice_server*.