Towards Leveraging Closed Captions for News Retrieval

Roi Blanco
Yahoo! Research
Barcelona, Spain
roi@yahoo-inc.com

Gianmarco De Francisci Morales
Yahoo! Research
Barcelona, Spain
gdfm@yahoo-inc.com

Fabrizio Silvestri
Yahoo! Research, Barcelona
ISTI - CNR, Pisa, Italy
f.silvestri@isti.cnr.it

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering, Query formulation

Keywords

intonow, news retrieval, continuous retrieval

1. INTRODUCTION

IntoNow[1] from Yahoo! is a second screen application that enhances the way of watching TV programs. The application uses audio from the TV set to recognize the program being watched, and provides several services for different use cases. For instance, while watching a football game on TV it can show statistics about the teams playing, or show the title of the song performed by a contestant in a talent show. The additional content provided by IntoNow is a mix of editorially curated and automatically selected one.

From a research perspective, one of the most interesting and challenging use cases addressed by IntoNow is related to news programs (newscasts). When a user is watching a newscast, IntoNow detects it and starts showing online news articles from the Web. This work presents a preliminary study of this problem, i.e., to find an online news article that matches the piece of news discussed in the newscast currently airing on TV, and display it in real-time.[2]

1.1 Problem Definition and Our Proposal

The problem is an interesting instance of matching different data sources, with a compelling real-world application. In order to analyze the content of the newscast we use the *closed captions* (CC) broadcasted along with it as its textual representation. There are several challenges involved in the design of an effective solution to the problem. First, we need to surface an article that is speaking exactly about the news currently airing among the thousands published every day. Second, the language used in TV and in news articles has different characteristics, and the text in the CC tends to be noisy, lacks proper capitalization and contains many typos and misspellings. Finally, article retrieval must happen as soon as possible in order to be valuable to the user.

In this work we propose a solution based on techniques from the realm of Information Retrieval (IR).

We break down the task into two sub-tasks: (i) find a good segmentation of the stream of CC, and (ii) retrieve relevant news as soon as possible. We model the newscast as a series of contiguous segments, each pertaining to a single cohesive topic. The *segmentation* problem consists in finding the boundaries of these news segments in the stream of CC. The *retrieval* problem consists in formulating a query given a segment, and issuing the query to an underlying news retrieval engine.

More formally, we are given an unbounded stream $\mathcal{C}$ of CC text that is divided into lines composed by a timestamp and a short piece of text. The timestamp increases monotonically in the stream, and represents the time at which the CC text is available to our system. Each line can be mapped to a topically-coherent segment S, which is however not available to our system. In addition, we are given in input a collection of documents, which can have any arbitrary format. We only assume the collection can be indexed and searched via an underlying IR engine. The problem is to find for each CC segment the most *appropriate* k news that match the topic discussed in the current newscast, as soon as possible.

Henzinger et al. [1] study the problem of finding online news articles *relevant* to the news currently being broadcasted, and also use closed captions to generate queries to a news search engine. However, we not only aim at being accurate, but we also take into account timeliness. Furthermore, we only consider news that match *exactly* the topic of each segment, rather then being only partly relevant. Finally, we introduce new quality metrics tailored to the problem.

In order to employ traditional IR techniques, both for solving the problem and for evaluation, we aim at finding ranked lists of documents. We identify three different components in our solution. First, the system identifies topics, i.e., segments of consecutive lines in $\mathcal{C}$ that belong to a cohesive theme. Second, it constructs a query out of the segment, which needs to represent the topic effectively, and needs to be matched against the document collection. In this work we use a keyword representation, but more generally the query could be comprised of different units, possibly capturing higher order semantics (e.g., named entities). Last, the system ranks documents in the collection for the query.

Given that news items have to be displayed as soon as possible, it is important to produce a query that can retrieve matching documents for the segment while being minimal, i.e., there is no other shorter prefix of words that is able to retrieve more matching documents.

[1] http://www.intonow.com

[2] This paper does *not* describe the design of the system currently in production at IntoNow.

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

We build a dataset containing both CC data coming from newscasts aired during 24 hours, and a repository of news articles from Yahoo! News spanning the two weeks preceding the date of the airing. We manually segment the caption into topically-coherent parts, and we ask expert evaluators to judge pairs of matching segments and articles. An article is judged a positive match only if it speaks about the *exact* same news as the closed caption segment.

2. QUALITY ASSESSMENT

The problem defined above bears some resemblance to information filtering, recommender systems and traditional information retrieval. In fact, the result of our system is a ranked list of news likely matching the one currently announced by the speaker of the newscast. Furthermore, as we already highlighted in the previous sections, time plays an important role in our application scenario.

To correctly evaluate the system, we need to take into account two conflicting goals: on the one hand we want to provide news items relevant for the topic of the current segment, on the other hand we want to provide these matchings as soon as possible. Providing results sooner means having less data at disposal to create a query for the topic of the current segment, which in turn can introduce noise and degrade relevance performance. Conversely, providing relevant results only when the current CC segment is over is of little value to the user of the application since by then the topic has already changed. The evaluation function needs to capture this trade-off.

Time-based relevance. We let the value of a match for a single segment depends on two factors: its relevance and the duration for which it is displayed on the screen of the user. For this reason, we define the relevance of a news match for a CC segment S as follows:

$$\phi(S) = \sum_{i=0}^{N} \nu\left(\mathcal{R}_i^k, S\right) \psi_\Gamma(t_i)$$

where $\mathcal{R}_i^k$ is i-th results list provided by our system, and t_i is the time at which it is provided. The function $\nu(\cdot)$ measures the *value* of a single ranked list of k documents $\mathcal{R}^k$ for the segment S, independently of time. In addition, we make use of a *time discount* function $\psi_\Gamma(t)$ to capture the notion that a match given at an earlier time is more valuable than the same match given at a later time. We experiment with four different time discount functions: $\mathit{Step} : \psi_\Gamma(t) = \mathrm{sgn}(\Gamma - t)$; $\mathit{Linear} : \psi_\Gamma(t) = 1 - {}^{t}/_{\Gamma}$; $\mathit{Logarithmic} : \psi_\Gamma(t) = 1 - \log_\Gamma(1 + t \times \frac{\Gamma-1}{\Gamma})$; $\mathit{Exponential} : \psi_\Gamma(t) = e^{-t\,{}^{10}/_{\Gamma}}$, where Γ is the length of the current CC segment.

To assess results, we use Mean Average Precision (MAP) and Precision at K (P@K) for small values of K as the measures for the value function $\nu(\cdot)$.

3. PRELIMINARY EXPERIMENTS

We perform preliminary experiments to assess the feasibility of our approach. We explore two simple approaches to segment the caption stream: sliding window (SW) and tumbling window (TW). Given a buffer $\mathcal{B}$ from a stream $\mathcal{C}$, and a window size in seconds Γ, SW$_\Gamma$ trims the oldest line from $\mathcal{B}$ when its size exceeds Γ and builds a new candidate segment of maximum duration Γ for each new line in the stream. TW$_\Gamma$ builds adjacent windows of fixed size Γ, proposing a new candidate segment and cleans $\mathcal{B}$ whenever adding a line to $\mathcal{B}$ would make it exceed Γ. The latter approach is the one used by Henzinger et al. [1].

Table 1: MAP scores normalized to the ORACLE score.

Variant	Step	Linear	Log.	Exp.
SW$_{10}$-TF-IDF	0.364	0.195	0.036	0.043
SW$_{30}$-TF-IDF	0.500	0.251	0.042	0.048
SW$_{60}$-TF-IDF	0.542	0.253	0.039	0.040
TW$_{10}$-TF-IDF	0.339	0.185	0.033	0.041
TW$_{30}$-TF-IDF	0.391	0.208	0.038	0.037
TW$_{60}$-TF-IDF	0.382	0.195	0.040	0.032

Table 2: P@1,3,5 scores normalized to the ORACLE scores, discounted with the Step function.

Method	P@1	P@3	P@5
SW$_{10}$-TF-IDF	0.617	0.528	0.466
SW$_{30}$-TF-IDF	0.761	0.672	0.613
SW$_{60}$-TF-IDF	0.774	0.715	0.657
TW$_{10}$-TF-IDF	0.581	0.492	0.433
TW$_{30}$-TF-IDF	0.617	0.543	0.493
TW$_{60}$-TF-IDF	0.539	0.515	0.478

We then use TF-IDF to select the most relevant terms in the current window. We experiment with different sizes for the window (10, 30 and 60 seconds), and compare the behavior of the four time discount functions. We normalize scores to an ORACLE which has access to the segment boundaries and uses the same TF-IDF technique to build queries.

Two clear insights can be drawn from the experiments in Tables 1 and 2. First, the SW approach always performs better than TW. Second, 30 seconds are a sweet spot for the window size. Smaller windows do not allow an effective representation of the topic, and larger windows lag behind topic changes. Notice how the proposed evaluation functions capture different aspects of the system. In Table 1, while the step function increases with Γ, the exponential one decreases. Indeed, the step function does not penalize late matchings so a larger window only brings benefits. Contrarily, the exponential function heavily penalizes lateness, thus imposing a tradeoff between accuracy and timeliness.

4. CONCLUSIONS

We performed an initial study on matching online news articles with a stream of closed captions from a newscast. We informally stated the problem and explored ideas on how to tackle it. Our study mainly borrowed techniques from IR and in our setting the main challenge is to match news articles with portions of closed captions as quickly as possible. Indeed, if we have the whole segment of text about a given topic, the news matching will be very precise, yet provide little value to the user. In this setting, we found that the sliding window approach performs better than the tumbling window one previously proposed in the literature.

5. REFERENCES

[1] M. Henzinger, B.-W. Chang, B. Milch, and S. Brin. Query-free news search. In *WWW '03: 12th International Conference on World Wide Web*, pp. 1–10, 2003.