Query Session Detection as a Cascade

Extended Abstract

Matthias Hagen, Benno Stein, and Tino Rüb

Bauhaus-Universität Weimar
<first name>.<last name>@uni-weimar.de

Abstract We propose a cascading method for query session detection in search engine logs (i.e., for finding consecutive queries a user submitted for the same information need). Our approach involves different detection steps that form a cascade in the sense that computationally costly features are applied only after cheap features "failed." This cascade is different to previous session detection approaches most of which involve many features simultaneously. Our experiments show the cascading method to save runtime compared to the state of the art while the detected sessions' accuracy is improved.

1 Introduction

We tackle the problem of query session detection from search engine logs. Detecting such search sessions is of major interest as they offer the possibility to support users stuck in longer sessions or to learn from the query reformulation patterns. Since the queries of one user in a log can be easily sorted by submission time, session detection is often modeled as the task of determining for each pair of chronologically consecutive queries whether the two queries were submitted for the same user information need.

We propose a new approach in form of a cascading method. Unlike former approaches to the problem the cascading method does not require the simultaneous evaluation of all features. Instead, it processes the features in different steps one after the other by increasing computational costs. Whenever a "cheap" feature allows for a reliable decision, features with higher cost are not computed. If, however, the cheaper features cannot reliably decide, more features are computed. For example, if a query contains the preceding query (e.g., `istanbul` and `istanbul archaeology`), it is reasonable to assume that both queries belong to the same session. No other feature besides a simple term overlap is needed for that decision. For more complex situations like `istanbul archeology` and `constantinople`, simple term overlap fails but computationally more costly features that are able to identify semantic similarities can support the desired decision of also assigning these two queries to the same session.

A use case of session detection is the extraction of sessions from a stored log in order to obtain sessions on which improved retrieval techniques can be evaluated. For this purpose, many studies in the literature use a basic time threshold since it is both easy to implement and very fast compared to more sophisticated methods developed in the session detection literature. Our cascading method aims at closing the gap between developed session detection methods and their application in practice: the cascading method is only about 5 times slower than a simple time threshold check but comes with a by far more reliable session accuracy.

2 Related Work

In a recent survey, Gayo-Avello compares most of the existing session detection approaches against a gold standard of human annotated queries [3]. At the same place he introduces the geometric method and shows its superiority over the existing methods in terms of detection accuracy. The geometric method involves only basic features and thus is very efficient. Also other approaches achieve convincing accuracy results in Gayo-Avello's study but come at the cost of evaluating many features for every pair of consecutive queries, leading to a bad runtime.

Session detection is often used as a pre-processing step to extract sessions from a query log on which, in turn, particular retrieval techniques are tested. In this regard many authors decide not to use methods developed in the session detection community but resort to some time threshold (like 10 or 15 minutes) between consecutive queries. Obviously, this is an easy to implement and fast way to detect sessions from a query log. But the resulting session's quality is not convincing [3].

Because of its efficiency the geometric method could close the gap between research on session detection methods and their application in practice. However, as a stand-alone approach, the geometric method (by design) is not able to detect semantic similarities of queries. Our cascading method builds upon the geometric method and introduces additional steps that are able to detect semantic similarities. Our objective is threefold: a runtime performance comparable to the geometric method, a similar ease of implementation, and a further improved session accuracy.

3 The Cascade: Step by Step

This section describes the steps of our cascading session detection method. From step to step the required features get more expensive but later steps are invoked only if the previous steps were not able to come to a reliable decision.

Throughout our explanations we view a query q as a set of keywords. For each query the search engine log additionally contains a user ID and a time stamp. If the user clicked on a result, the log also contains the clicked result's rank and URL. We assume that the queries of one user in the log are ordered by submission time and explain our cascading framework for the queries submitted by one user. We model the problem of session detection as the problem of deciding for each consecutive pair q, q' of queries whether the session s that contains q continues with q' or whether q' starts a new session.

Step 1: Simple Query String Comparison

The most simple patterns that two consecutive queries q and q' may form can be detected by a simple comparison of the keywords: repetition ($q = q'$), generalization ($q' \subseteq q$), and specialization ($q \subseteq q'$). Whenever two consecutive queries represent one of these three cases, our approach assigns them to the same session regardless of the time that has passed between their submission. The rationale is that in case of a longer time between a repetition, generalization, or specialization pattern we assume the user to have continued a pending session. While Step 1 is able to reliably detect session continuations for repetitions, generalizations or specializations, it would decide "new session" in all other cases, which is not always correct. For these other cases our

cascading method invokes the geometric method [3] in Step 2 as a more sophisticated comparison of the query strings.

Step 2: Geometric Method

The geometric methods relaxes Step 1's query overlap condition with respect to the elapsed time between the queries. This way, session continuations can be detected for query pairs that are syntactically more different than simple repetition, generalization, or specialization patterns.

Let t and t' be the submission times of a pair q and q' of consecutive queries. Using the offset $t' - t$, the geometric method computes the time feature $f_{\text{time}} = \max\{0, 1 - \frac{t'-t}{24h}\}$. Thus, chronologically very close queries achieve scores near to 1 whereas longer time periods between query submissions decrease the score until it gets 0 for queries with a gap of 24h or larger. The syntactic similarity for q' is computed as the cosine similarity $f_{\cos}$ between the character 3- to 5-grams of the query q' and the session s whose current last query is q. The geometric method votes for a session continuation iff $\sqrt{(f_{\text{time}})^2 + (f_{\cos})^2} \geq 1$. This decision rule can be geometrically interpreted as plotting the point $(f_{\text{time}}, f_{\cos})$ in the $\mathbb{R}^2$ and checking whether it lies inside or outside the unit circle.

Although the geometric method detects queries with overlapping terms more reliably than does Step 1, there are problematic cases where the geometric method should not be trusted. Such cases include query pairs where f_{time} is large (i.e., chronologically close queries) but the syntactic similarity reflected by $f_{\cos}$ is rather low. An example is the pair `istanbul archeology` and `constantinople` from the introduction. Assuming that the session started with `istanbul archeology`, the only overlapping 3- to 5-grams are `sta`, `stan`, and `tan` but nevertheless one would expect both queries to belong to the same session as semantically they are very similar. In pilot experiments we determined that for query pairs with $f_{\cos} < 0.4$ and $f_{\text{time}} > 0.8$ (queries that are chronologically close but that have a small n-gram overlap) the geometric method's decision misses many semantically similar queries and wrongly assigns them to different sessions. Hence, for query pairs that fall in this range, our cascading method drops the geometric method's decision and invokes Step 3 to further analyze semantic similarity of the current query pair.

Step 3: Explicit Semantic Analysis

An elegant way to compare semantic similarity of two texts is the explicit semantic analysis (ESA) introduced by Gabrilovich and Markovitch [2]. The idea is to not compare the given two texts directly but to use an index collection against which similarities are calculated. Since the index collection (e.g., the Wikipedia articles) can be preprocessed and stored, invoking ESA is not too expensive compared to the basic session detection features such as n-gram overlap or query submission time.

The ESA principle works as follows. During a preprocessing step, a $tf{\cdot}idf$-weighted term-document-matrix of the Wikipedia articles is stored as the ESA matrix. During runtime, the two to-be-compared texts represented as vectors are multiplied with the

ESA matrix and the cosine similarity of the resulting vectors yields the ESA similarity. In our setting, the two texts that should be ESA-compared are the keywords of q' and all the keywords of the queries in the session s to which the previous query q belongs. As Anderka and Stein [1] showed that the ESA accuracy varies only very little with the size of the index collection, we conducted a pilot experiment with different numbers of Wikipedia articles and finally used a sample of 100 000 as the ESA index collection.

One problem of ESA applied to queries is that the texts that are compared are rather short. Hence, to have a reliable decision, we only use ESA to detect session continuations and choose an ESA similarity threshold of 0.35 that has to be achieved as an argument for a session continuation. Given the case that the ESA similarity is below the threshold, we do not immediately view q' as the start of a new session but view ESA's decision as "not sure" and invoke Step 4 of the cascade that aims at enlarging the representation of the queries that are compared.

Step 4: Search Result Comparison

Step 4 uses the web search results of the queries q and q'. Since retrieving these results requires index accesses at search engine site or the submission of two (time consuming) web queries from an external client, the web results are applied only if all previous steps failed to provide a reliable decision.

Using web search results to detect semantically similar queries is not a new idea (cf. [4] for example) but is applied in different variants. Some authors use the URLs of the retrieved documents, others fetch the complete documents. Moreover, different numbers of search results are used in the literature, ranging from the top-10 documents up to the top-50. We evaluated different settings in a pilot study and finally chose to compare the sets of URLs of the top-10 retrieved documents via the Jaccard coefficient (ratio of common top-10 URLs of q and q'). Whenever the Jaccard coefficient is at least 0.1 (i.e., q' returns at least one of the top-10 results of q), we view this as an argument for a session continuation. Otherwise, q' is treated as the start of a new session.

4 Experimental Evaluation

To ensure comparability, we evaluated our cascading method on the annotated gold standard query corpus that Gayo-Avello used in his experiments [3]. The corpus contains 11 484 queries of 223 users sampled from the AOL query log. The queries are manually subdivided into 4 254 sessions with an average of 2.70 queries per session. For evaluation purposes we use the F-Measure $F_\beta = \frac{(1+\beta^2)\cdot prec \cdot rec}{\beta^2 \cdot prec + rec}$, where precision and recall for the detected sessions are measured against the human gold standard. We follow Gayo-Avello and set $\beta = 1.5$, which emphasizes wrong session continuations as the bigger problem compared to wrong session breaks. Gayo-Avello reports an F-Measure of 0.9184 for his geometric method and we could verify his results in our experiments. The cascading method improves upon this value and achieves an F-Measure of 0.9323.

Our experiments reveal that Step 2 requires about 2.25 times more time for a query pair analysis than Step 1. Hence, on the about 40% of the queries that Step 1 detects as repetitions, generalizations, or specializations, our cascade saves time compared to the

geometric method (also note that for these queries, Step 1 always correctly votes for a session continuation). After Step 2, about 75% of the queries are reliably judged (F-Measure of 0.9184), such that Step 3, which requires about 1.08 times more time than Step 2 with a preprocessed ESA matrix in main memory, is invoked on only 25% of the queries. Hence, after Step 3, our cascading approach is still faster than the original geometric method. The only crucial issue for runtime is Step 4, which we implemented against the Bing API and which requires more than 20 times the runtime of Step 1. Step 4 is invoked on about 22% of the queries but increases the F-Measure only slightly: after Step 3 we already achieve 0.9315. I.e., when efficiency is an issue, Step 4 can be omitted, still having both an improved session accuracy compared to the original geometric method and improved runtime. However, also note that at search engine site Step 4 can be operationalized at much higher efficiency,

Since a potential use case of our method is session extraction from a stored log file (e.g., in a pre-processing to obtain sessions on which some improved retrieval techniques should be evaluated), we also suggest a very fast second version of our method (without Step 4) that assigns a "not sure, maybe new session" decision when Step 3 votes for a new session. A post-processing can remove all sessions involved in a "not sure" decision (this removes about 22% of all the queries). The remaining sessions then achieve an F-Measure of 0.9755. I.e., this version of the cascading method with post-processing can be used as a very fast and reliable session extraction method that produces high quality sessions resembling the ones a human would have extracted.

5 Conclusion and Outlook

We have presented a cascading session detection approach, based on the geometric method and the well-known ESA retrieval model, which is able to achieve very high query session detection accuracy against a human gold standard. Our method sensibly invokes time consuming features only when cheaper features failed to provide a reliable session detection. Equipped with a post-processing step that drops sessions with "not sure" decisions, the accuracy of our approach is almost perfect on Gayo-Avello's gold standard. Hence, the system could be applied as a pre-processing for many evaluations that need to extract high quality sessions from query logs as experimental data.

An interesting aspect for future research is to invoke a post-processing that is able to account for multitasking at user site (cf. [4]). The goal then is to merge sessions into a hierarchy that resembles different levels of search goals and missions.

Bibliography

[1] M. Anderka and B. Stein. The ESA retrieval model revisited. In *Proceedings of SIGIR 2009*, pages 670–671.

[2] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI 2007*, pages 1606–1611.

[3] D. Gayo-Avello. A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences*, 179(12):1822–1843, 2009.

[4] R. Jones and K.L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of CIKM 2008*, pages 699–708.