Segmenting User Sessions and Missions in Search Engine Query Logs by Leveraging Word Embeddings

Pedro Gomes · Bruno Martins · Luís Cruz

Abstract Segmenting user interactions as registered in search engine query logs, according to the underlying information needs (e.g., delimiting user sessions), is important to perceive information needs and assess how they are satisfied, to enhance the quality of search engine rankings, and to better direct content to certain users. Most previous methods use human judgments to inform supervised learning algorithms, and/or use global thresholds on temporal proximity and on simple lexical similarity metrics. This paper presents an unsupervised method for segmenting user sessions that improves on the current state-of-art, leveraging additional heuristics and similarity metrics derived from word embeddings. We specifically extend a previous approach based on combining temporal and lexical similarity measurements, integrating semantic similarity components that use pre-trained FastText embeddings. Building on the session segmentation method, the paper also advances an unsupervised approach for detecting search missions (i.e., sets of queries, not necessarily continuous, referring to the same information need inside a multitasking behavior pattern, and/or considering hierarchical goals). We report on experiments with two different subsets from the well-known AOL query dataset, both used in previous studies. The first subset contains a total of 10,235 queries, with 4,253 sessions, 2.4 queries per session, and 215 unique users, being used to evaluate the effectiveness of our algorithm for detecting sessions. The second subset was used to evaluate the effectiveness of our algorithm for detecting search sessions and missions, containing a total of 8,840 queries, with 2,881 sessions, 1,378 missions, 3.1 queries per session, 6.42 queries per mission, and 127 unique users. The results attest to the effectiveness of the proposed methods, which outperform a large set of baselines also corresponding to unsupervised techniques.

Keywords Analysis of Search Engine Query Logs · User Session Detection · User Mission Detection · String Similarity Metrics · Word Embeddings

Pedro Gomes
INESC-ID and Instituto Superior Técnico, Universidade de Lisboa, Portugal
E-mail: pedro.almeida.gomes@tecnico.ulisboa.pt

Bruno Martins
INESC-ID and Instituto Superior Técnico, Universidade de Lisboa, Portugal
E-mail: bruno.g.martins@tecnico.ulisboa.pt

Luís Cruz
INESC-ID and Faculdade de Engenharia, Universidade do Porto, Portugal
E-mail: luiscruz@fe.up.pt

1 Introduction

In the context of user interactions with search engines, the notion of session is critical to the study of user habits and intentions when using these systems. In brief, a session is a sequence of activities followed by one individual to satisfy an information need, regardless of the elapsed time, number of interactions with the system, or the existence of interruptions on these interactions. Identifying user sessions is important to understand a search engine's effectiveness in suggesting content pointers to user's searches, with several previous studies suggesting that by studying the properties of these sessions (e.g., clicks and dwell-time on search results) one can evaluate system quality and predict general user satisfaction [4, 12, 16, 18, 22, 21].

According to Hagen et al. [11], there are two possible scenarios where the identification of user sessions can have a beneficial result: online, where it can help the search engine to present better results or to suggest queries that other users submitted in similar situations;

and offline, where the identification of sessions within information collected from query logs gives information about the behavior of users, supporting the evaluation of their satisfaction. Typically, a user session ends when the user has satisfied his information need, or when the user decided to stop (i.e., the user leaves the system or proceeds to a new information need). The correct identification of user search sessions is still a challenging problem, which entails the following main issues:

- Session boundaries can be highly ambiguous. The most commonly used approach to identify user sessions involves grouping all interactions from the same user that happened within a constrained temporal interval. This approach is simple and efficient, having been reported to achieve a confidence level of approximately 70 percent on Web search logs [17]. However, this approach also introduces noise, since it does not take into account activity breaks or sessions that are very long. One of the difficulties in using a global temporal threshold is that true session intervals usually have a smooth distribution, and it is almost guaranteed that longer sessions will be handled incorrectly by setting a temporal threshold. Nonetheless, some previous proposals attempt to address this issue through variable thresholds depending on the user [23].
- It can be difficult to use query terms to infer if two queries from the same user belong to the same session, e.g. when they do not have search terms in common but correspond to the same underlying topic. For example, if one user searches for *iPhone* and then searches for *Apple*, there is no direct way to assess the underlying similarity between these query terms, although recent proposals have addressed the issue through semantic representations for the terms employed in user queries (e.g., through distributional semantic models and word embeddings methods [2,8]).
- Query logs often do not feature unique user identifiers, instead containing only cookie identifiers in a fraction of the records, plus information on source IP addresses and user-agents (i.e., identifiers for the type of Web browser in which the query was submitted). In these cases, the logs may feature queries from different users (and consequently also from different sessions) appearing interleaved in chronological order, all associated to the same IP address (e.g., from a common Internet proxy). Leveraging user-agent information together with IP addresses can partially address this problem, and approaches that extend the notion of session in order to better handle activity breaks can also be beneficial.
- It can be challenging to identify multitasking behavior or hierarchical sub-tasks with different purposes, where the related interactions will not appear as continuous in the search log. For instance, when planning vacations users will likely perform different queries corresponding to booking flights or hotels. In these cases, the queries can perhaps be grouped according to different sessions, even though they relate to the same underlying task or encompassing mission. The queries relating to a same sub-task may not appear continuous in the query log, and considering semantic similarity will not be enough to properly group queries according to search missions.
- Efficiency is a major concern, either in the context of timely online identification of user sessions, or in the processing of very large query logs for segmenting session and mission boundaries. Depending on the circumstances, there is usually a trade-off between algorithm accuracy and efficiency.

Tackling the aforementioned challenges usually involves a combination of different and carefully selected heuristics: temporal approaches (e.g., relying on a global threshold) have the advantage of simplicity and efficiency, although they also have problems with accuracy, whereas heuristics capturing lexical/topical similarity (e.g., methods based on distributional semantics) can have a high accuracy, although also a lower computational efficiency. By combining these general methods, we aim at achieving a useful trade-off between effectiveness and efficiency.

In this article, extending on a previous paper presented at TPDL'19 [8] by considering both user search sessions and missions, we start by describing and evaluating an unsupervised method for segmenting user sessions that improves upon a previous proposal by Gayo-Avello [7], which is based on a geometric interpretation for how temporal and lexical similarity measurements can be effectively combined.

For comparing consecutive queries from the same user, our approach uses a geometric method to decide if the more recent query belongs to the same session of the first, combining a user-specific temporal threshold with a lexical similarity value computed from n-gram overlaps. If a reliable decision cannot be made with this procedure, we compute separate semantic similarity measurements, e.g. with basis on pre-trained FastText embeddings [2]. The FastText approach represents tokens based on their character n-grams, and is capable of producing representations for query terms that were not seen during model training.

Additionally, building on the session segmentation method, we also present an unsupervised approach for segmenting search missions, taking into consideration

multitasking behavior and hierarchical goals for consecutive and non-consecutive user sessions. For efficiently assessing if sessions from the same user belong to the same underlying search mission, our approach first relies on an improved version of the geometric method, comparing the last query q of session s and the first query q' of session s'. If the decision is not reliable, we then use a semantic similarity step based on pre-trained FastText embeddings [2].

We evaluated both methods against several alternative approaches, using two different subsets of the well-known 2006 AOL search query log [25] that, despite the many privacy concerns[1], has frequently supported studies on user session identification. The entire dataset has 30 million queries from 650,000 different users, collected over a period of three months. From the entire dataset, Gayo-Avello [7] built a subset with ground-truth annotations for user search sessions, containing 10,235 queries from 215 different users, where 4,253 sessions were manually identified. In turn, Hagen et al. [9] assembled a different subset with ground-truth annotations for user sessions and search missions. This subset contains a total of 8,840 queries, with 2,881 sessions, 1,378 missions, 3.1 queries per session, 6.42 queries per mission, and 127 unique users.

On the AOL data subset from Gayo-Avello, our method to identify user sessions achieved a precision of 88.19, a recall of 95.13, and an F1-score of 91.53, which is about 2.96% better than the unsupervised geometric method advanced by Gayo-Avello [7]. Ablation tests also confirmed the usefulness of the different components involved in the proposed method. On the other hand, on the AOL data subset from Hagen et al. [9], without fine-tuning of the parameters we achieved a precision of 84.49, a recall of 92.29, and an F1-score of 88.22 when identifying user sessions, proving that regardless of dataset our method outperforms state-of-art approaches. Finally, our method to identify search missions was also evaluated on the subset of the AOL query log from Hagen et al., achieving a B^3 precision of 79.73, a B^3 recall of 70.61, and an F1-score of 74.90. As a side contribution, we deliver a reproducibility package[2] with the datasets plus all the scripts used in the evaluation experiments that are reported on this article.

The rest of the article is organized as follows: Section 2 surveys previous work in the area, while Section 3 details the proposed approaches for session segmentation and mission identification. Section 4 presents the experimental evaluation of the proposed methods, detailing the datasets, the evaluation methodologies, and the obtained results. Finally, Section 5 summarizes our main conclusions and presents possible directions for future work in the area.

2 Related Work

Previous studies addressing user session identification have proposed methods (i) based only on temporal thresholds (i.e., the time gap between queries), (ii) based only on lexical heuristics (i.e., string similarity between queries, search patterns, etc.), and (iii) based on a combination of both these types of heuristics, either relying on supervised learning methods or instead using completely unsupervised approaches.

The most common methods to identify user sessions are based on a global temporal threshold, in which two consecutive queries belong to the same session if the elapsed time is less than a pre-defined threshold. Previous studies have considered limits of 5 [3], 10-15 [13], or 30 minutes [26]. This approach is still widely used in practice (e.g., Google claims to apply a threshold of 30 minutes in their Web analytics application[3]), but an important limitation relates to the application of the same threshold for all contexts (e.g., different users may behave differently, and different query logs may reflect particular system and user characteristics that affect the typical duration of the sessions).

Mehrzadi and Feitelson [23] proposed an approach which deals with the aforementioned limitation, adapting the temporal threshold based on the activity of each user. The proposed approach leverages temporal gaps between consecutive queries and binning on a logarithmic scale. First, for each user, the authors extract the time gap between all consecutive queries, and then they create a histogram with basis on these values. The bins in the histogram are built based on powers of 2, starting with the interval from 0 to 32 seconds. The authors also restrict the bins to be analyzed in the subsequent steps, ranging from 512 seconds in the lower value to 8192 seconds in the upper value. Each candidate bin is finally scored, based on how much lower its count value is than the maximal count value on its two sides. The upper value for the bin with a higher score is chosen as the user threshold, to be used when segmenting user sessions. In case of a tie, the upper value from the bin closest to 1,200 seconds is chosen.

Instead of relying on temporal thresholds, other studies have proposed to use lexical similarity heuristics. For instance, Bernard et al. [15] proposed an

[1] https://www.nytimes.com/2006/08/09/technology/09aol.html

[2] https://github.com/PedroG1515/Segmenting-User-Sessions

[3] https://support.google.com/analytics/answer/2731565

approach to detect session boundaries based on query reformulations, considering that two queries do not belong to the same session if they do not have query terms in common. Lucchese et al. [20] proposed, among other approaches, a clustering method that leverages the Jaccard similarity coefficient [14] computed from character 3-grams, to assess the similarity between queries. Despite yielding a high accuracy, these methods are slower than temporal thresholds and can also introduce several problems, namely (i) they are limited to sub-string matches that cannot detect semantic similarity between queries, and (ii) they directly assign similar queries to the same user session, independently of the time interval separating these queries.

Considering the aforementioned limitations, Gayo-Avello [7] proposed a method that combines temporal and lexical heuristics. The temporal component f_t can be calculated as shown in Equation 1 [11], where t_i and t_{i+1} correspond to the timestamps for the consecutive user queries being assessed:

$$f_t = \max\left\{0, 1 - \frac{t_{i+1} - t_i}{24 \text{ hours}}\right\} \quad (1)$$

The lexical component f_l is based on representing queries as sets of character 3-grams and assessing the overlap between these representations. To decide if two queries belong to the same session, the author proposed a geometric interpretation for how temporal and lexical similarity measurements, both in the interval $[0, 1]$, can be combined. The method corresponds to computing the area enclosed by positive semi-axes and a unit circle centered at $(1, 1)$, as shown in Equation 2:

$$\sqrt{f_t^2 + f_l^2} \geq 1 \quad (2)$$

Hagen et al. [11] proposed a cascade method (i.e., an incremental procedure based on a sequence of heuristics), which also combines temporal and lexical components. This method relies first on more efficient features (i.e., query re-formulation patterns), and then progressively on more complex features with higher effectiveness and lower efficiency, only using the more complex features if strictly necessary to obtain reliable results. Query reformulation patterns (e.g., query repetition, as well as query generalization or specialization through the inclusion/removal of terms) are first used to detect the similarity of two consecutive queries, regardless of the time between them. Although this step is very efficient, it has a low efficiency because it does not detect misspellings or other vocabulary mismatches. The second step uses the aforementioned geometric method from Gayo-Avello [7] to refine the results, being invoked only when Step 1 decided for a new session (i.e., when no query repetition or re-formulation was detected). In this case, the authors used Equation 1 to compute the temporal component, but the lexical component was instead based on the cosine similarity between vector representations encoding 3- to 5-grams. The third step uses Explicit Semantic Analysis (ESA) as a refinement over lexical similarity [5], capturing the semantic similarity between the new query and all the keywords of the queries in the session to which the previous query belongs. ESA is applied on the cases having a high temporal similarity (i.e., greater than 0.8) but a low lexical similarity (i.e., less than 0.4), thus being incorrectly classified according to the geometric method. ESA does not compare representations for the set of terms under analysis directly, instead building representations from a background collection of documents (e.g., a large random sample of Wikipedia articles) so that each term is represented as a column vector in the TF-IDF matrix of the background corpus, and a query (i.e., a set of terms) is represented as the centroid of the vectors representing the terms. The ESA vectors are compared through the cosine similarity and if the result is greater than a predefined threshold (i.e., 0.35 in the experiments reported by Hagen et al. [11]), the two queries belong to the same session. In the final step, which is computationally more demanding, the authors use Web search results to detect semantically similar queries, comparing the top retrieved documents for two consecutive queries. If there is at least one URL in common in the sets of top retrieved documents, then the queries are considered to be in the same user session.

The authors compared the cascade method against the geometric method from Gayo-Avello [7], obtaining improvements in terms of recall and in the F1-score, although also a lower precision. In general, the cascade method is very reliable, although the final step is an important bottleneck in terms of performance. The authors concluded that it may be preferable to ignore the last step, this way achieving a better trade-off between efficiency and effectiveness.

In more recent work, Hagen et al. [9] proposed an improved version of the cascade method, adding two additional steps. In the first additional step, which is executed in the beginning, the authors used a global threshold of 90 minutes between consecutive queries to improve efficiency. All sequences of queries from the same user that are separated by less of 90 minutes are then analyzed through the other steps in the cascade.

Between the Explicit Semantic Analysis (ESA) and the URL similarity steps, the authors also considered an additional step in the cascade, which used Linked Open Data (LOD) as a refinement to detect semantic similarity, identifying DBpedia entities in the user

queries through a previously proposed query segmentation algorithm [10]. First, the main entities of consecutive queries are identified from Wikipedia titles, and they are mapped to their corresponding DBpedia entities via a dictionary that unifies different expressions into the same generic entity in the LOD graph (e.g., *John Fitzgerald Kennedy* and *JFK*). Considering nodes corresponding to Wikipedia entities and edges corresponding to references between entities, for each entity node e the authors consider a list that contains all entities that are directly connected to entity e. Each entity e has an idf-inspired weight given by $\log\left(\frac{pl}{pl_e}\right)$, where pl is the total number of lists (i.e., entity nodes) and pl_e is the total number of lists that contain entity e. The *idf* weights are also normalized according to $\frac{idf_e}{idf_{max}}$, where *max* is the entity with largest *idf* value.

The LOD step identifies all one- and two-step paths in the resulting graph, to assess the minimum length of paths between entities in the queries with at most one intermediate entity (i.e., paths from one main entity from query q to one main entity from query q'). The authors calculate the maximum weight associated to these paths, in which the weight of a one-step path is equal to one, and the weight of a two-step path is equal to the *idf* weight of the intermediate node. If the maximum is greater than 0.6 the queries are marked as corresponding to the same user session.

Hagen et al. [9] also introduced a new algorithm to detect search missions, which merges different sessions from the same user by applying the cascade approach. The authors essentially followed the same algorithm defined for identifying sessions, with the following changes for detecting missions: (i) the steps where time is involved as a feature were removed, and (ii) instead of comparing all queries for each session the authors only compared the last query from the first session with the first query from the second session, to efficiently enable comparing non-consecutive user sessions.

Hagen et al. [9] compared the new version of the cascade method against the geometric method from Gayo-Avello [7], and also against the original cascade method [11], obtaining improvements in terms of the F1-score and in efficiency for the detection of sessions. Interestingly, the original cascade method [11] only achieves an F1-score of 85.30 on a new subset of the AOL query log that was introduced in this subsequent study [9], which is quite lower than the value reported in the original study (i.e., and F1-score of 93.27). Although both datasets were obtained from the 2006 AOL query log [25], they contain different users and behaviors. In terms of mission detection, the cascade method correctly identified 865 of the 1134 mission continuations, with errors in 307 sessions due to semantic mismatches.

Previous studies have also advanced supervised methods for session segmentation, in some cases leveraging user activity during a search (e.g., clicks or dwell time on search results) for extending existing lexical and time features. For instance, Ozmutlu et al. [24] proposed a method based on thresholding the results from a linear regression with two-factor interactions, using features corresponding to search patterns, time between consecutive queries, and the sequential position of the query within the session. Jones and Klinkner [17] proposed to learn a binary classifier for inferring whether two queries belong to the same task, leveraging temporal features, lexical similarity features based on words or characters, query co-occurrence features, and features derived from the search results. Despite the interesting results, supervised approaches require training data, thus being harder to generalize to new application domains. We aim to detect sessions and missions in the query logs of different systems, using suitable heuristics that operate directly on the logs, and avoiding the supervised training of a classifier tailored to these tasks.

3 Unsupervised Segmentation of Search Sessions and Search Missions

To address the problems of segmenting user sessions and missions from user interactions registered in search engine query logs, we propose a new unsupervised approach combining multiple heuristics, evaluating it against a set of baselines that covers the current state-of-the-art. An ablation analysis was also considered, checking the impact of temporal, lexical, and semantic similarity heuristics, in the overall methods that integrate them. We first describe the individual heuristics that were considered for the ablation tests, and which are also the main components of the complete method. Then, we describe the complete method for segmenting user sessions, detailing some of the components that are involved (e.g., the use of word embeddings). Finally, we describe an extension to our unsupervised approach, which addresses the problem of identifying search missions.

3.1 Individual Heuristics and Ablated Approaches for the Segmentation of User Sessions

In terms of temporal heuristics, our tests with baselines and ablated models for search session segmentation considered two different approaches. The first re-

lies on a global temporal threshold, in which two consecutive queries belong to the same user session if the elapsed time is less than a pre-defined threshold. With basis on previous studies, we tested the standard values of 5 [3], 15 [13], and 30 minutes [26]. The second approach considers a user-specific temporal threshold, defined with basis on the user distribution of intervals between consecutive queries, as originally proposed by Mehrzadi and Feitelson [23].

In terms of lexical and semantic heuristics, our ablation tests considered three different approaches. The first is based on the Jaccard similarity coefficient between sets of 3- and 4-grams extracted from the new query, and from all the queries in the session of the previous query. The second is based on pre-trained FastText embeddings [2], computing the cosine similarity between averaged embedding vectors for the words present in the consecutive queries. Finally, the third approach also relies on FastText embeddings, but in this case we use the word mover's distance [19] to assess the similarity between sets of embeddings, respectively for words in the new query, and for words in all the queries in the session of the previous query. We used an existing implementation[4] for computing the earth mover's distance between word embeddings. In all three approaches, we tested different thresholds (i.e., between 0.1 and 0.9) in the obtained similarity value. The lexical similarity computations did not involve language-specific lists of stop-words or stemming algorithms, although we ignored punctuation symbols and sub-strings such as `www.` or `.com` (i.e., navigational queries often contain URLs, and we ignored common URL tokens from the string similarity computations).

In terms of combinations for multiple heuristics, besides our complete approach, we also tested three different baseline methods. The first was the geometric method from Gayo-Avello [7], with the same parameters proposed by the author. The second baseline corresponds to an improved version of the geometric method, which instead of using the overlap between character 3-grams, in the lexical component, uses the Jaccard similarity coefficient between sets of character n-grams of lengths 3 and 4, extracted from the last query and from all the queries in the session of the previous query. In the improved version of the geometric method, we also changed the temporal component in Equation 1, using a normalization constant equal to the minimum between 24 hours or twice the maximum time between consecutive queries for the user under analysis, instead of the fixed normalization constant of 24 hours. Finally, the third baseline method corresponds to a slightly different procedure from the complete method described in Section 3.2, using only the cosine similarity between averaged word embeddings for consecutive queries, instead of using the word mover's distance.

In the methods combining multiple heuristics that use a threshold over the Jaccard similarity coefficient between character n-grams (although not on the lexical baselines that use the Jaccard coefficient alone), we used a simple two-step approach to improve the computational performance, based on the intuition that a fast lower-bound for the similarity can be computed from the length of common prefixes and/or suffixes. First, notice that for a non-empty string of size k, the maximum number of distinct n-grams is given by $\max(1, k-(n-1))$. For two strings in which one is a prefix or a suffix of the other (i.e., strings resulting from a typical reformulation pattern, corresponding to the addition or removal of terms from the search query), in which the size of the common sub-string is k_1 and the length of the longer string is k_2, the number of n-grams in common cannot be higher than $\max(1, k_1-(n-1))$, and the number of distinct n-grams cannot be lower than the number of n-grams in common, or higher than $\max(1, k_2-(n-1))$. The ratio between these two quantities gives us an approximation on the Jaccard similarity coefficient, that we can use as a lower-bound. This procedure will, in some cases, lead to wrong lower-bound estimates when there are many n-grams appearing repeated in the strings. However, in such cases, the strings under comparison will still have a significant match in their contents, and we can use the estimate in a way that is similar to the query reformulation patterns in the cascade method [11].

In the combined methods, when checking if the Jaccard similarity coefficient is above a given threshold, we first check if the lower-bound (computed with basis on the similarity towards the last query) is greater than the threshold, and only if this is not enough to reach a decision do we compute the actual similarity coefficient. Notice that Equation 2 from the geometric method corresponds to a minimum threshold of $\sqrt{1-f_t^2}$ on the (lower-bound to the) Jaccard similarity.

3.2 The Complete Approach for Session Segmentation

Our complete method for segmenting user sessions corresponds to a cascade approach, extending the geometric method and the approaches from Hagen et al. [11, 9] through the use of FastText word embeddings. The complete method consists of the following three steps, and it can also be summarized through the pseudo-code that is shown in Algorithm 1.

[4] `https://radimrehurek.com/gensim/models/keyedvectors.html#gensim.models.keyedvectors.WordEmbeddingsKeyedVectors.wmdistance`

Algorithm 1 The Proposed Search Session Segmentation Method

1: Sort the log that registers user interactions using the userID as a first criterion, and then using the timestamp
2: Initialize each query in the log as belonging to a separate search session
3: **for each** user u **do**
4: $t_{\max_u}$ = Maximum time between consecutive queries for user u
5: **for each** pair of consecutive sessions i and $i+1$ from the same user u **do**
6: $f_t = \max\left\{0, 1 - \frac{t_{i+1} - t_i}{\min\{24 \text{ hours}, 2 \times t_{\max_u}\}}\right\}$
7: f_{l_1} = Lower-bound on the Jaccard similarity coefficient
8: **if** $f_{l_1} > \sqrt{1 - f_t^2}$ **then**
9: Merge the two consecutive sessions into a single one
10: **else**
11: f_{l_2} = Jaccard similarity based on character n-grams with the parameter $n \in \{3, 4\}$
12: **if** $\sqrt{f_t^2 + f_{l_2}^2} > 1$ **then**
13: Merge the two consecutive sessions into a single one
14: **else**
15: **if** $f_t > 0.7$ and $f_{l_2} < 0.5$ **then**
16: f_{s_1} = Cosine similarity from averaged word embeddings
17: **if** $f_{s_1} > 0.5$ **then**
18: Merge the two consecutive sessions into a single one
19: **else**
20: f_{s_2} = Word mover's distance [19] from sets of word embeddings
21: **if** $f_{s_2} < 0.1$ **then**
22: Merge the two consecutive sessions into a single one
23: **else**
24: **if** $\sqrt{f_{s_1}^2 + (1.0 - f_{s_2}^2)} > 1$ **then**
25: f_u = Similarity from largest common sub-strings in the strings corresponding to URLs
26: **if** $f_u > 0.7$ **then**
27: Merge the two consecutive sessions into a single one

First, we use the aforementioned improved version of the geometric method, relying on a per-user maximum threshold for the temporal component, using the fast lower-bound on the Jaccard similarity coefficient, and using character 3- and 4-grams in the lexical component. A new query will belong to the session of the previous query if an adapted version of Equation 2 (i.e., using the condition greater than one, instead of greater or equal to one, thus ensuring that equal queries separated by very large time spans are not merged) is satisfied, and otherwise different sessions will be considered. Although effective on its own, this step fails at capturing semantic similarities (i.e., it incorrectly classifies many cases involving queries in close temporal proximity, but with a low lexical similarity). Adapting the cascade method from Hagen et al. [11,9], if the temporal proximity is above the threshold value of 0.7, and the lexical similarity is below the threshold of 0.5, we attempt to merge the queries into the same session according to the results of the second step.

In the second step, starting on Line 16 from Algorithm 1, we use pre-trained FastText embeddings [2] to quickly assess the semantic similarity of two consecutive queries/sessions. The FastText approach, which involves training vector representations through a language modeling task that involves predicting words occurring in the same context, as illustrated in Figure 1,

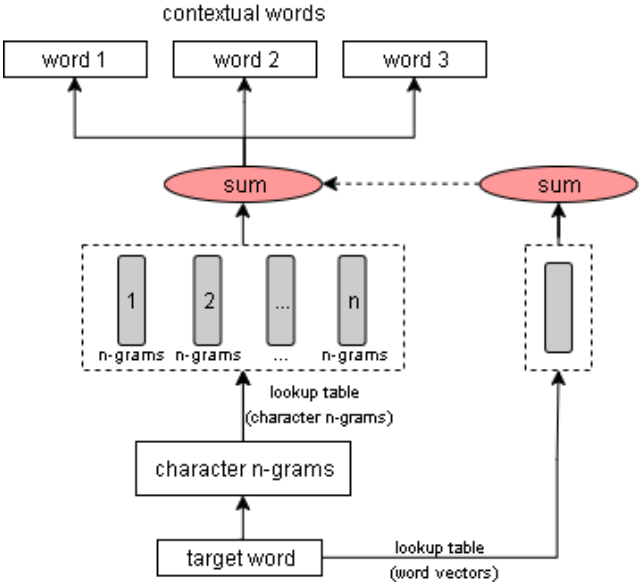


Fig. 1 Overview on the training of FastText embeddings.

has been shown to perform well in representing words (e.g., it has been used on models for sentence classification as a down-stream task leveraging word embeddings), especially in the case of rare words, by making use of character level information. Each word is seen as a bag of character n-grams, in addition to the word it-

self. During model training, FastText learns weights for each of the n-grams, as well as for the entire word tokens. Rare words can be properly represented, since it is highly likely that some of their n-grams also appear in other words, and even out-of-vocabulary words can be represented, by taking the average of the embeddings for the corresponding n-grams.

We start by measuring the cosine similarity between averaged word embeddings for the consecutive queries. If the similarity is above the threshold of 0.5, then the queries are defined to belong to the same session. Otherwise, we compare the set of embeddings for the words in the query, against the set of embeddings corresponding to words in all the queries belonging to the session of the previous query. In this second case, a fast algorithm for computing the word mover's distance [19] is used to compare the sets of embeddings and, if the resulting distance is lower than 0.1, we assume that the two consecutive queries belong to the same session (otherwise different sessions will be considered). The Word Mover's Distance (WMD) is a special case of the well-studied earth mover's distance transportation problem, measuring the dissimilarity between two sets of embeddings as the minimum amount of distance that the embeddings of one set need to travel to reach the embeddings of another set.

In the third step, if the decision remains unreliable (i.e., if the WMD distance is too large, but the cosine and WMD similarities from the previous step are within particular thresholds, also according to a geometric interpretation) we will compare clicked URLs through the longest sub-strings in common. We assume that the query log under analysis contains information on clicked URLs (i.e., for each query, if the user accessed one of the URLs in the search results, then the corresponding URL is registered on the log together with the query). We first normalize the URLs by removing redundant information, including prefixes corresponding to protocol specifications (e.g., `http://` or `https://`), sub-strings corresponding to top-level domain names (e.g., `.com`, `.org` or `.edu`), or suffixes corresponding to popular file extensions (e.g., `.html`, `.jsp` or `.php`). Then, we compute the longest common sub-strings between any of the normalized URLs associated with the new query, and any of the normalized URLs associated to queries in the same session as the previous query. If any of these longest common sub-strings has a length that is at least 70% of the length of one of the URLs for the new query, then the queries are considered to belong to the same user session, and otherwise a different user session will be considered for each of the search queries.

Tables 1.A to 1.D illustrate the obtained results at the different steps of the algorithm, for a sample of queries taken from the AOL query log (i.e., a sequence of queries for the same user, that made system interactions generally related to the topic of music). Table 1.A shows the segmentation boundaries made by a human annotator, whereas Tables 1.B to 1.D show the boundaries resulting from each step of the algorithm, progressively reconstructing the same decisions as the human annotator. For instance, Table 1.C shows that by considering semantic similarity based on word embeddings, one can almost reconstruct the first session from Table 1.A. Step 3 effectively refines the results by leveraging URLs, although the method still failed at joining the last two iterations on the table. In the example that is presented, the user clicked at most in one URL for each query.

3.3 The Proposed Approach for Grouping Queries According to Search Missions

In the context of query log analysis, and in addition to user sessions, considering search missions can provide a more general view of usage patterns. Search missions should group user sessions in a way that encodes (i) multitasking behavior (i.e., we should identify non continuous interactions that are broken across different sessions, but that refer to the same underlying task or more general information need) and (ii) hierarchical sub-tasks with different purposes, but that contribute to achieving the same more general goal. For instance, when booking a vacation, many people check out restaurants, flights, and hotels. All these different queries may be grouped in different user sessions, that nonetheless relate to the same underlying mission.

To identify user search missions, we propose an extension to the unsupervised method from the previous section, in which Algorithm 1 is considered as the first step. Thus, assuming that the queries are already grouped into user sessions, through Algorithm 1, we will leverage a separate and similar cascade approach to merge user sessions into search missions.

Contrary to what has been reported by other studies, we consider that the temporal component is a valid feature for identifying search missions, since similar goals, even if not continuous, will likely be close in time. Nevertheless, the application of the same threshold for all contexts can be noisy, since different users may behave differently in each situation.

Our approach analyzes pairs of sessions s and s' belonging to the same user, not necessarily contiguous, using an adapted version of the method to identify user sessions, that in the first steps only considers the last query q of s and the first query q' of a subsequent session s'. A perhaps more reasonable approach would involve

Table 1 Results for a sample of queries taken from the subset of the 2006 AOL query log hat was released by Gayo-Avello [7].

A. Segmentation from Human Annotator

Query	URL	Time
teeth like god's shoeshine lyrics	www.selyrics.com	1142351220
grills lyrics		1142369580
grills lyrics nelly	www.lyrics07.com	1142369580
blink 182 lyrics	www.azlyrics.com	1142371620
edit the sad parts lyrics	www.selyrics.com	1142372820
my lips are cold the truth is told lyrics	www.lyricsdepot.com	1142375940
the authority song		1142449620
the authority song lyrics	www.selyrics.com	1142449680
black dresses by spill canvas	www.azlyrics.com	1142449980
playing for keeps lyrics	www.lyrics07.com	1142450040
rhyming dictionary	www.rhymer.com	1142452560
monstr in a wheelchair		1142465880

B. Segmentation Resulting from Step 1

Query	URL	Time
teeth like god's shoeshine lyrics	www.selyrics.com	1142351220
grills lyrics		1142369580
grills lyrics nelly	www.lyrics07.com	1142369580
blink 182 lyrics	www.azlyrics.com	1142371620
edit the sad parts lyrics	www.selyrics.com	1142372820
my lips are cold the truth is told lyrics	www.lyricsdepot.com	1142375940
the authority song		1142449620
the authority song lyrics	www.selyrics.com	1142449680
black dresses by spill canvas	www.azlyrics.com	1142449980
playing for keeps lyrics	www.lyrics07.com	1142450040
rhyming dictionary	www.rhymer.com	1142452560
monstr in a wheelchair		1142465880

C. Segmentation Resulting from Step 2

Query	URL	Time
teeth like god's shoeshine lyrics	www.selyrics.com	1142351220
grills lyrics		1142369580
grills lyrics nelly	www.lyrics07.com	1142369580
blink 182 lyrics	www.azlyrics.com	1142371620
edit the sad parts lyrics	www.selyrics.com	1142372820
my lips are cold the truth is told lyrics	www.lyricsdepot.com	1142375940
the authority song		1142449620
the authority song lyrics	www.selyrics.com	1142449680
black dresses by spill canvas	www.azlyrics.com	1142449980
playing for keeps lyrics	www.lyrics07.com	1142450040
rhyming dictionary	www.rhymer.com	1142452560
monstr in a wheelchair		1142465880

D. Segmentation Resulting from Step 3

Query	URL	Time
teeth like god's shoeshine lyrics	www.selyrics.com	1142351220
grills lyrics		1142369580
grills lyrics nelly	www.lyrics07.com	1142369580
blink 182 lyrics	www.azlyrics.com	1142371620
edit the sad parts lyrics	www.selyrics.com	1142372820
my lips are cold the truth is told lyrics	www.lyricsdepot.com	1142375940
the authority song		1142449620
the authority song lyrics	www.selyrics.com	1142449680
black dresses by spill canvas	www.azlyrics.com	1142449980
playing for keeps lyrics	www.lyrics07.com	1142450040
rhyming dictionary	www.rhymer.com	1142452560
monstr in a wheelchair		1142465880

always comparing query q' against all queries from s, or even comparing all queries from s against all queries from s'. However, the proposed approach offers a better overall comprise between algorithm accuracy and computational efficiency.

As a first step for grouping search sessions into missions, we use the improved version of the geometric method, considering only the last query q of s and the first query q' of s' on the lexical component, and last timestamp t of s together with the first timestamp t' of s' on the temporal component. However, instead of using a global threshold of 24 hours as in Equation 1, we propose to use a more relaxed threshold of 48 hours.

In the second step, to capture semantic similarity, we adapted the cascade method from Hagen et al. [11, 9]. If the temporal similarity is greater than 0.5 and the lexical similarity is less than 0.7, we calculate the cosine similarity between average embedding vectors for queries q and q'. If the returned value from the cosine similarly is greater than 0.5, sessions s and s' will be grouped in the same search mission. If the decision is still unreliable (i.e., for similarities below 0.5), we calculate the word mover's distance between sets of embedding vectors for all queries in session s, and all queries from session s'. If the value is less than 0.3, sessions s and s' will be grouped in the same search mission.

Finally, in a third step, we compare the clicked URLs associated to queries from s and q', through the longest sub-strings in common, as described in the previous section. If the longest common sub-string has a length that is at least 70% of the length of one of the URLs for the more recent query q', then the sessions s and s' will be grouped in the same mission. Notice that, instead of using all URLs from s', we are only considering URLs from query q', which can nonetheless be more than one.

If we consider the example from Table 1.A, there are two sessions and only one search mission, since all interactions can be said to correspond to the same encompassing information need related to music lyrics. The proposed method would correctly identify the single search mission that is present in the example.

4 Experimental Evaluation

This section describes the experimental evaluation of the proposed methods. We first present a statistical characterization of the AOL datasets that supported our tests, together with the considered experimental methodology. Then, Subsection 4.2 presents and discusses the obtained results.

4.1 Datasets and Experimental Methodology

The datasets used in our experiments correspond to two subsets of the AOL query log released on August 2006 [25]. To ensure a meaningful comparison against previously published results for session segmentation, we used the subsets of the AOL query log, with ground-truth annotations regarding sessions, that were respectively made available by Gayo-Avello [7] and by Hagen et al. [9]. Both datasets were used in several previous studies in the area. The subset from Gayo-Avello has a total of 10,235 queries from 215 unique users, which are divided into 4,253 sessions with an average of 2.4

queries per session. The subset made available by Hagen et al. [9] also contains ground-truth annotations for user missions, and it contains a total of 8,840 queries, with 2,881 sessions, 1,378 missions, 3.1 queries per session, 6.42 queries per mission, and 127 unique users. This second subset was used in our experiments to evaluate both session and mission segmentation.

In both datasets, each record contains the following attributes: (i) userID (i.e., a unique user identifier); (ii) query text (i.e., the set of keywords submitted by the user); (iii) URL (i.e., the URL that the user clicked after receiving the results for the query, or empty if no clicks were made); (iv) timestamp (i.e., the instant when the user submitted the query); and (v) session boundary (i.e., a Boolean indicator for whether the query marks the beginning of a new session, according to the ground-truth annotations). The dataset made available by Hagen et al. [9] has the following additional attributes: (i) click rank (i.e., the position in the list of search results for the document being clicked), and (ii) mission boundary (i.e., a unique search mission identifier). In both methods, the records (i.e., the user queries) are first sorted according to userID (i.e., joining together queries from the same user), and then sorted according to the timestamp, prior to analysis.

To better understand each dataset, we first looked at consecutive queries from the same users, judged by the human annotators as belonging or not to the same search session or mission. The distributions for several characteristics associated to these consecutive queries are depicted in Figure 2, which shows side-by-side the distribution for (a) the temporal proximity in minutes, (b) the lexical similarity according to the Jaccard coefficient between character n-grams (i.e., 3- and 4-grams) from consecutive queries, and (c) the semantic similarity according to the cosine similarity between averaged word embeddings of consecutive queries. The figure shows that all three heuristics have different distributions for the consecutive queries in each of the four classes (i.e., same versus different sessions, and same versus different missions, respectively), although the three heuristics seem to capture different cases. Through our experiments, we attempted to assess the contribution of each heuristic.

To evaluate the proposed approach for session segmentation, we relied on the same evaluation methodology and metrics considered by Gayo-Avello [7], corresponding to notions of precision and recall. Precision is defined as the ratio between the number of consecutive queries for which there is a change of session where the algorithm has agreed with the ground-truth, and the number of consecutive queries for which the algorithm predicted a change of session. Recall, on the other hand, is defined as the ratio between the number of consecutive queries for which there is a change of session where the algorithm agreed with the ground-truth, and the number of consecutive queries corresponding to a session change in the ground truth.

On the other hand, to evaluate our proposed approach for identifying search missions, we used the B^3 clustering metrics discussed by Enrique Amigó [1]. The mission groupings generated by the proposed algorithm are seen as clusters of queries, which we compare against ground-truth clusterings corresponding to the missions that are present in the annotated dataset.

Being $\mathrm{R}(q)$ and $\mathrm{P}(q)$ respectively the real and predicted mission groups for a query q, we can define the correctness of the relation between queries q and q' as:

$$\mathrm{Correctness}(q, q') = \begin{cases} 1 & \text{if } \mathrm{R}(q) = \mathrm{R}(q') \longleftrightarrow \mathrm{P}(q) = \mathrm{P}(q') \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The B^3 precision of a query is the proportion of correctly related queries in the predicted mission group for the query (including the query itself), and the overall B^3 precision is the averaged precision of all queries. The B^3 recall is analogous, replacing the predicted mission group by the true mission group. An F1-Score can also be computed as usual, i.e. by taking the harmonic mean of B^3 precision and recall.

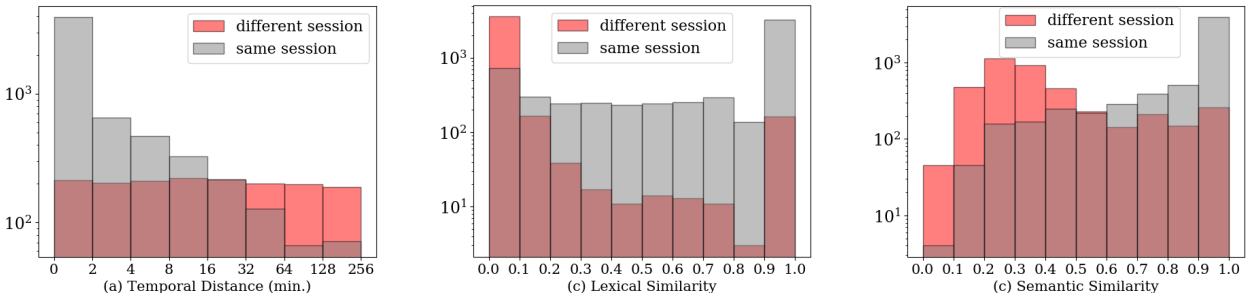
Besides assessing the quality of the predictions, we also measured the time involved in processing the entire subsets of the AOL query log. The measurements for the different methods were all made in a standard PC with an Intel Core i7 8700K (3.7 GHz) CPU, an SSD drive where the log file was stored, and 64 Gb of RAM.

4.2 Experimental Results

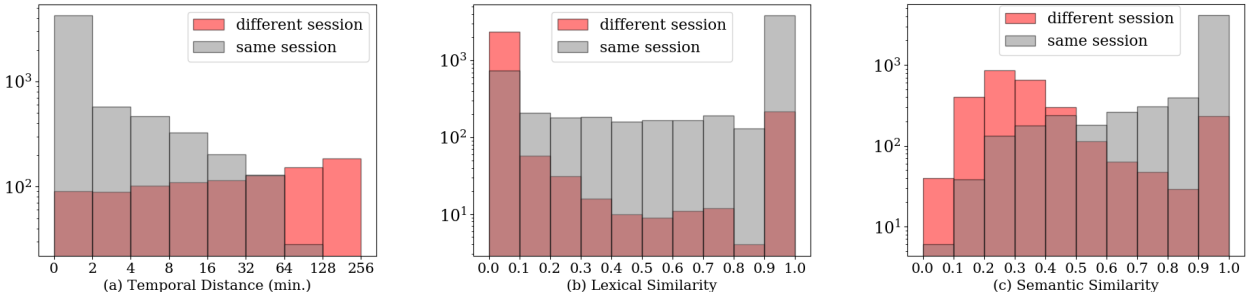
We compared the different baseline approaches for session segmentation that are listed in Section 3.1, against the complete method given in Section 3.2. Table 2 presents the obtained results for segmenting sessions, both over (a) the subset of the AOL query log from Gayo-Avello [7], and (b) the subset of the 2006 AOL query log from Hagen et al. [9]. The table shows that the complete procedure outperforms all the considered baselines in terms of the F1-score, although also with a higher computation time.

The results illustrate that methods based on a global temporal threshold already achieve a very satisfactory performance for session segmentation (i.e., F1-scores of 82.40 and of 87.81, in the datasets from Gayo-Avello [7] and Hagen et al. [9], when using thresholds of 15 and

Distributions for the number of queries according to search sessions in the dataset from Gayo-Avello [7].



Distributions for the number of queries according to search sessions in the dataset from Hagen et al. [9].



Distributions for the number of queries according to search missions in the dataset from Hagen et al. [9].

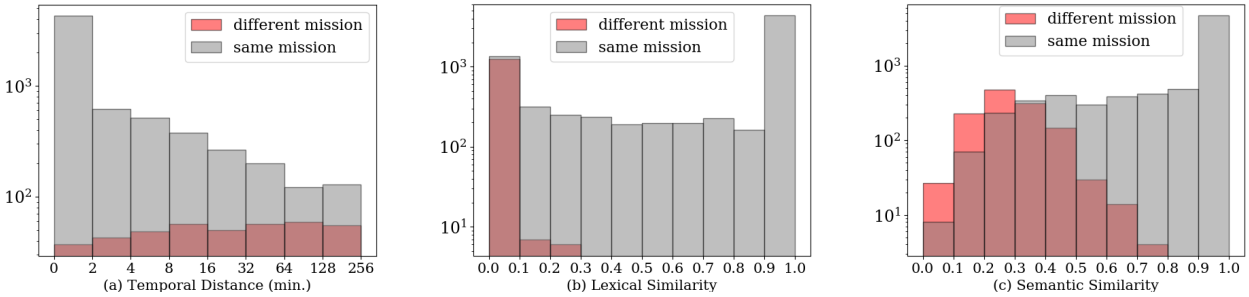


Fig. 2 Comparison of the distribution for the number of consecutive queries, according to (a) temporal proximity/distance, (b) lexical similarity, and (c) semantic similarity. The top row corresponds to search sessions in the subset that was released by Gayo-Avello [7], the middle row corresponds to search sessions in the subset that was released by Hagen et al. [9], and the bottom row corresponds to the search missions also in the subset released by Hagen et al. [9]

30 minutes, respectively), at the same time also being faster. Relying on user-specific temporal thresholds is not much slower, although we failed to outperform the results obtained with a global threshold. When using a lexical similarity heuristic alone, the results over the dataset from Gayo-Avello [7] are slightly better than those obtained with a temporal threshold, although the semantic similarity heuristics (i.e., both methods relying on word embeddings) alone perform slightly worse. On the dataset from Hagen et al. [9], the results with the lexical and semantic heuristics alone are clearly worse than those achieved with temporal thresholds. Moreover, the lexical and semantic approaches are also much slower compared to temporal approaches.

In terms of the combined methods, the proposed approach outperforms all individual baselines in both subsets, the simpler geometric method, and the variants that were considered, although at the cost of higher computation time. The improved geometric method, and a variation of the proposed method that does not use the word mover's distance, both offer a good compromise between result quality and computation time.

Table 3 shows the results for mission identification over the subset of the 2006 AOL query log proposed by Hagen et al. [9], confirming that the complete approach to identify search missions also outperforms the baselines in terms of the F1-score. The different rows in Table 3 correspond to alternative heuristics (or sequences of heuristics) for detecting search missions, in

Table 2 Performance metrics for different user session segmentation methods (a) over the subset of the 2006 AOL query log that was released by Gayo-Avello [7], and (b) over the subset of the 2006 AOL query log that was released by Hagen et al. [9].

Component	Method		Gayo-Avello [7]			Hagen et al. [9]			Execution Time (m.sec)
			Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Temporal	Global Threshold	$T = 5$	77.00	87.54	81.93	75.21	91.91	82.72	1615
		$T = 15$	84.91	80.04	82.40	87.40	86.43	86.91	
		$T = 30$	89.00	75.12	81.47	93.70	82.61	87.81	
	Threshold per User		90.68	71.15	79.74	**96.37**	79.35	87.04	1975
Lexical or Semantic	Jaccard Coefficient	>= 0.1	83.67	92.50	87.86	77.24	87.16	81.90	3090
		>= 0.3	75.42	94.59	83.93	69.38	90.35	78.49	
		>= 0.5	69.49	95.18	80.33	63.94	91.15	75.16	
		>= 0.7	64.26	95.67	76.88	59.34	91.84	72.10	
		>= 0.9	60.62	96.10	74.34	55.56	92.40	69.40	
	Word Embeddings Cosine	>= 0.1	**95.87**	7.64	14.16	95.29	6.32	11.85	4090
		>= 0.3	90.58	54.03	67.69	88.91	49.81	63.85	
		>= 0.5	84.82	88.53	86.63	80.07	82.99	81.51	
		>= 0.7	76.05	93.98	84.07	71.22	89.17	79.19	
		>= 0.9	65.16	95.65	77.52	60.32	91.81	72.80	
	Word Embeddings WMD	<= 0.1	61.49	95.79	74.90	56.04	92.29	69.74	6049
		<= 0.3	64.82	92.23	76.47	59.03	91.46	71.75	
		<= 0.5	69.45	92.55	79.35	63.80	90.91	74.98	
		<= 0.7	74.42	90.24	81.57	68.76	89.55	77.79	
		<= 0.9	79.25	87.11	83.00	73.72	86.78	79.72	
Temporal + Lexical and Semantic	Geometric Method (GM)		88.24	88.90	88.57	83.16	87.75	85.39	3542
	Improved GM		83.93	**97.60**	90.25	79.38	**95.28**	86.61	2287
	Proposed Method		88.06	95.20	91.49	84.37	92.36	88.19	4983
	Proposed Method (WMD)		88.19	95.13	**91.53**	84.49	92.29	**88.22**	5771

Table 3 Performance metrics for different user mission segmentation methods, as obtained over the subset of the 2006 AOL query log that was released by Hagen et al. [9].

Component	Method		Identified Sessions			Execution Time (m.sec)
			B^3Precision	B^3Recall	B^3F1-Score	
	No Heuristics (Use the Sessions)		83.49	65.41	73.35	5798
Temporal	Global Threshold	$T = 5$	81.03	68.12	74.01	6987
		$T = 15$	77.96	70.21	73.88	
		$T = 30$	75.86	70.69	73.19	
	Threshold per User		76.44	71.52	73.90	8024
Lexical or Semantic	Jaccard Coefficient	>= 0.1	72.92	72.81	72.86	9058
		>= 0.3	75.20	70.10	72.80	
		>= 0.5	78.00	70.16	73.87	
		>= 0.7	80.14	69.90	74.67	
		>= 0.9	79.14	69.01	73.73	
	Word Embeddings Cosine	>= 0.1	58.16	**79.03**	67.01	11587
		>= 0.3	67.61	74.69	70.97	
		>= 0.5	74.49	70.80	72.59	
		>= 0.7	76.27	70.80	73.44	
		>= 0.9	77.86	69.69	73.55	
	Word Embeddings WMD	<= 0.1	77.27	69.99	73.45	13587
		<= 0.3	77.27	69.45	73.15	
		<= 0.5	76.80	69.67	73.06	
		<= 0.7	76.16	69.93	72.91	
		<= 0.9	73.93	70.99	72.43	
Temporal + Lexical and Semantic	Geometric Method (GM)		80.94	68.89	74.43	9568
	Improved GM		**83.52**	65.35	73.32	8679
	Proposed Method		80.15	70.74	**75.15**	12693
	Proposed Method (All Queries)		77.51	68.72	72.85	19023

all cases relying on the best approach (i.e., the complete method that achieved the higher F1-score) for identifying search sessions. The column with the execution time refers to the time involved in segmenting search sessions, together with the time involved in grouping search sessions into missions. On average, more than 80% of the overall execution time relates to grouping search sessions into missions.

Methods based on a global temporal threshold already achieve good results (i.e., an F1-score of 74.01, when using a global threshold of 5 minutes), being also faster. Relying on user-specific temporal thresholds is not much slower, although this failed to outperform the results obtained with a global threshold. On the other hand, depending on the threshold, lexical heuristics alone can perform slightly better than the results obtained with temporal thresholds, while semantic heuristics alone are slightly worse. The task of user mission segmentation seems to be significantly harder than the segmentation of user sessions, and many of the considered baseline methods failed to outperform the results that would be obtained by simply considering

each user session as a different mission (i.e., the first line in Table 3). The proposed combination method achieves a slightly higher accuracy than the baseline unsupervised approaches, while not significantly expanding the computational effort.

Unexpectedly, the improved version of the geometric method does not achieve good results on its own. In the last method that is shown in Table 3, each step of our complete approach involves comparing all the queries from session s against all queries from session s', confirming that comparing all queries brings noise and is much slower than previous methods. Finally, the geometric method offers a good compromise between result quality and computation time.

Figure 3 further details the results for session segmentation in both datasets, plotting the variation on precision, recall, and the F1-score (a) for a baseline corresponding to a global temporal threshold, as a function of that threshold, and (b) for the complete method, as a function of the threshold (b.1) on the cosine similarity between sets of word embeddings, or (b.2) on the similarity between URLs in both approaches (i.e., user sessions and missions segmentation).

The results on the subset that was released by Gayo-Avello [7], which are shown in chart (a), confirm that a temporal threshold of approximately 15 minutes corresponds to the best trade-off in terms of the F1-score, while Charts (b.1) and (b.2) show that the thresholds that were given in Section 3.2, in connection to the proposed method, are also adequate. On the other hand, the results on the subset that was released by Hagen et al. [9] illustrate that each threshold could be better adjusted to this dataset. For instance, if we set the cosine similarity threshold to 0.4 we would achieve an F1-score of 88.76 for session segmentation.

5 Conclusions and Future Work

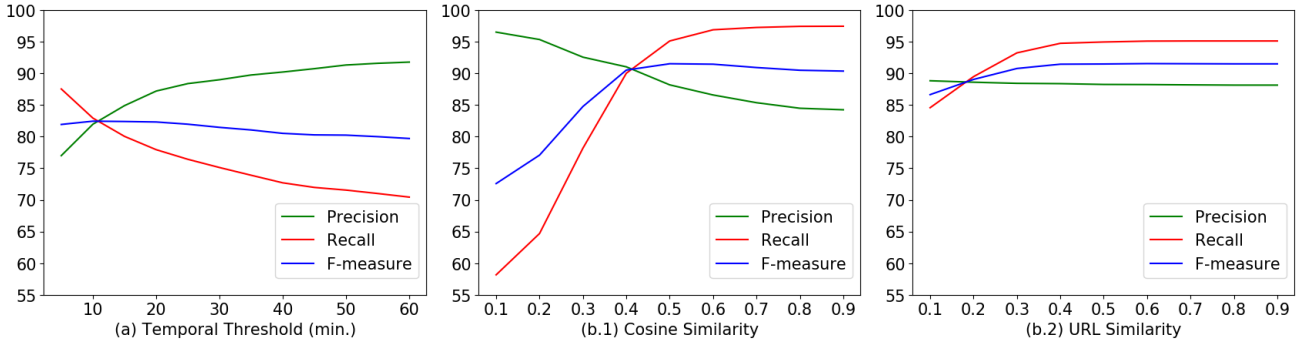
Segmenting user sessions and missions in search engine query logs are important tasks in the context of several applications. Both these tasks are nonetheless quite challenging, for instance involving the detection of query reformulations that employ a variety of factors: correcting misspellings (e.g., *gogle* versus *google*), using co-referential expressions (e.g., *CEO Facebook* versus *Mark Zuckerberg*), acronyms (e.g., *John Fitzgerald Kennedy* versus *JFK*), generalizations and specializations (e.g., *Jaws* versus *Jaws the movie*), etc.

This article presented new unsupervised procedures for session and mission segmentation, improving upon current state-of-the-art methods [7,11,9] through the usage of pre-trained word embeddings. Our experiments confirmed the effectiveness of the proposed methods, which achieve a higher segmentation accuracy than competing unsupervised approaches, while not significantly expanding on the computational effort. The performance of the proposed algorithms, together with the fact that they do not require training data or significant parameter tuning, makes them ideal for processing very large query logs from real-world search systems, independently of the domain (e.g., we plan to use this method to analyze the query log of a national search engine for legislative contents, in the context of an ongoing technology transfer project).

Despite the interesting results, there are also many possibilities for improvement in future work. For instance, instead of relying on traditional string similarity metrics (e.g., the Jaccard similarity coefficient computed between character n-grams), we can perhaps experiment with the use of learned similarity functions [28,27,6], which have been shown to achieve significantly better results in other types of string matching problems. One such similarity function could be pre-trained on general data from other domains besides query logs (e.g., on large collections of alternative names for Wikipedia entities), and then used in our segmentation procedures. Another idea relates to the inclusion of additional steps in the proposed procedures, improving on the computational performance by first relying on simpler string similarity metrics with a very permissive threshold, and later using increasingly more reliable, although more compute-intensive, string similarity functions with more restrictive thresholds.

One particular challenge that we plan to tackle in future work (i.e., in the ongoing project related to a search engine for legislative contents) relates to the fact that query logs often do not feature unique user identifiers, instead containing only cookie identifiers in a fraction of the records, plus information on source IP addresses and user-agents (i.e., identifiers for the type of Web browser in which the query was submitted). In these cases, the logs may feature queries from different users (and consequently also from different sessions) appearing interleaved in chronological order, all associated to the same IP address (e.g., from a common Internet proxy). We plan to adapt the proposed procedure for session segmentation in this specific scenario, first by sorting the query records according to the combination of user cookie, IP address, and user-agent (instead of sorting records according to userID, as in our experiments), and then by considering also a post-processing step similar to the approach for mission identification, merging different non-consecutive sessions (perhaps interleaved with interactions from different users sharing the same IP address and user agent) that are temporally and thematically coherent.

Variations in performance for session segmentation in the data subset from Gayo-Avello [7].



Variations in performance for session segmentation in the data subset from Hagen et al. [9].

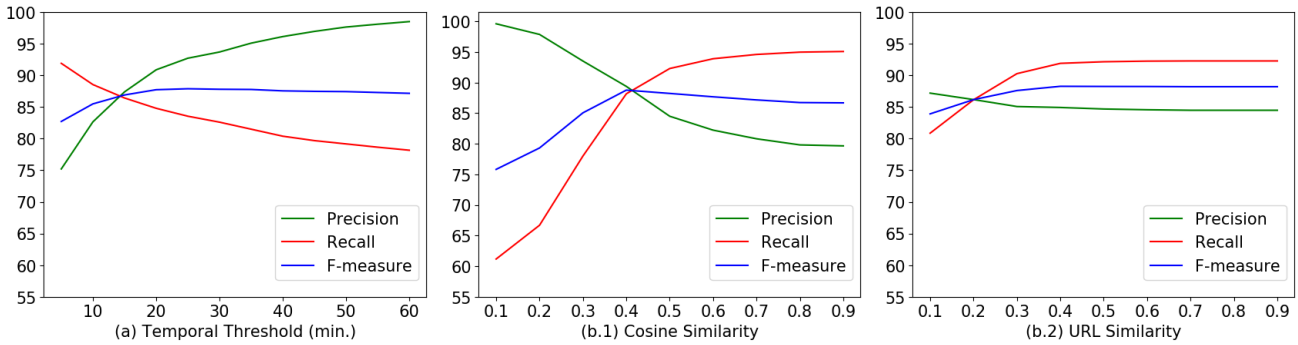


Fig. 3 Variations in precision, recall, and F1-scores for session segmentation, as a function of thresholds in both datasets.

Acknowledgements

This work was partially supported by Fundação para a Ciência e Tecnologia (FCT), through the GoLocal project with reference CMUP-ERI/TIC/0046/2014, and also through the INESC-ID multi-annual funding from the PIDDAC program (UID/CEC/50021/2019).

References

1. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval **12**(4) (2009)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5** (2017)
3. Downey, D., Dumais, S.T., Horvitz, E.: Models of searching and browsing: Languages, studies, and application. In: Proceedings of the International Joint Conference on Artificial Intelligence (2007)
4. Feild, H., Allan, J., Jones, R.: Predicting searcher frustration. In: Proceedings of the ACM Conference on Research and Development in Information Retrieval (2010)
5. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the International Joint Conference on Artificial Intelligence (2007)
6. Gan, Z., Singh, P.D., Joshi, A., He, X., Chen, J., Gao, J., Deng, L.: Character-level deep conflation for business data analytics. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (2017)
7. Gayo-Avello, D.: A survey on session detection methods in query logs and a proposal for future evaluation. Information Sciences **179**(12) (2009)
8. Gomes, P., Martins, B., Cruz, L.: Segmenting user sessions in search engine query logs leveraging word embeddings. In: International Conference on Theory and Practice of Digital Libraries (2019)
9. Hagen, M., Gomoll, J., Beyer, A., Stein, B.: From search session detection to search mission detection. In: Proceedings of the Conference on Open Research Areas in Information Retrieval (2013)
10. Hagen, M., Potthast, M., Beyer, A., Stein, B.: Towards optimum query segmentation: in doubt without. In: Proceedings of the International Conference on Information and Knowledge Management (2012)
11. Hagen, M., Stein, B., Rüb, T.: Query session detection as a cascade. In: Proceedings of the ACM Conference on Information and Knowledge Management (2011)
12. Hassan, A., Shi, X., Craswell, N., Ramsey, B.: Beyond clicks: Query reformulation as a predictor of search satisfaction. In: Proceedings of the ACM Conference on Information and Knowledge Management (2013)
13. He, D., Göker, A.: Detecting session boundaries from web user logs. In: Proceedings of the BCS-IRSG Annual Colloquium on Information Retrieval Research (2000)
14. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytologist **11**(2) (1912)
15. Jansen Bernard, J., Spink, A., Blakely, C., Koshman, S.: Defining a session on web search engines. Journal of the

American Society for Information Science and Technology **58**(6) (2007)
16. Jiang, J., Hassan Awadallah, A., Shi, X., White, R.W.: Understanding and predicting graded search satisfaction. In: Proceedings of the ACM Conference on Web Search and Data Mining (2015)
17. Jones, R., Klinkner, K.L.: Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In: Proceedings of the ACM Conference on Information and Knowledge Management (2008)
18. Kim, Y., Hassan, A., White, R.W., Zitouni, I.: Modeling dwell time to predict click-level satisfaction. In: Proceedings of the ACM Conference on Web Search and Data Mining (2014)
19. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: Proceedings of the International Conference on Machine Learning (2015)
20. Lucchese, C., Orlando, S., Perego, R., Silvestri, F., Tolomei, G.: Identifying task-based sessions in search engine query logs. In: Proceedings of the ACM Conference on Web Search and Data Mining (2011)
21. Mayr, P., Kacem, A.: A complete year of user retrieval sessions in a social sciences academic search engine. In: Proceedings of the International Conference on Theory and Practice of Digital Libraries (2017)
22. Mehrotra, R., Awadallah, A.H., Shokouhi, M., Yilmaz, E., Zitouni, I., El Kholy, A., Khabsa, M.: Deep sequential models for task satisfaction prediction. In: Proceedings of the ACM on Conference on Information and Knowledge Management (2017)
23. Mehrzadi, D., Feitelson, D.G.: On extracting session data from activity logs. In: Proceedings of the Annual International Systems and Storage Conference (2012)
24. Ozmutlu, S., Ozmutlu, H.C., Spink, A.: Automatic new topic identification in search engine transaction logs? Using multiple linear regression. In: Proceedings of the Hawaii International Conference on System Sciences (2008)
25. Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: Proceedings of the International Conference on Scalable Information Systems (2006)
26. Radlinski, F., Joachims, T.: Query chains: learning to rank from implicit feedback. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2005)
27. Santos, R., Murrieta-Flores, P., Calado, P., Martins, B.: Toponym matching through deep neural networks. International Journal of Geographical Information Science **32**(2) (2018)
28. Santos, R., Murrieta-Flores, P., Martins, B.: Learning to combine multiple string similarity metrics for effective toponym matching. International Journal of Digital Earth **11**(9) (2018)