PSemQE: Disambiguating Short Queries Through Personalised Semantic Query Expansion

Oliver Baumann[1][a] and Mirco Schoenfeld[1][b]
[1]*University of Bayreuth, Universitätsstraße 30, 95447 Bayreuth, Germany*
{*oliver.baumann,mirco.schoenfeld*}*@uni-bayreuth.de*

Keywords: Information Retrieval, Query Expansion, User-Centric Design, Word Embeddings, Word Sense Disambiguation

Abstract: Locating items in large information systems can be challenging, especially if the query has multiple senses referring to different items: depending on the context, *Amazon* may refer to the river, rainforest, or a mythical female warrior. We propose and study Personalised Semantic Query Expansion (PSemQE) as a means of disambiguating short, ambiguous queries in information retrieval systems. This study examines PSemQE's effectiveness in retrieving relevant documents matching intended senses of ambiguous terms and ranking them higher versus a base query without expansion. Synthetic user profiles focused on narrow domains were generated to model well-defined information needs. Word embeddings trained on these profiles were used to expand queries with semantically related terms. Experiments were conducted on corpora of varying sizes to measure the retrieval of predetermined target articles. Our results show that PSemQE successfully disambiguated polysemous queries and ranked the target articles higher than the base query. Furthermore, PSemQE produces result sets with higher relevance to user interests. Despite limitations like synthetic profiles and cold-start issues, this study shows PSemQE's potential as an effective query disambiguation engine. Overall, PSemQE can enhance search relevance and user experience by leveraging user information to provide meaningful responses to short, ambiguous queries.

1 INTRODUCTION

Large knowledge repositories have become a staple of digital life; however, navigating their content can be daunting for users. Galleries, libraries, archives, and museums consistently strive to make their collections accessible through digital catalogues; there are hardly any commodities that cannot be located in the vast catalogue of e-commerce platforms, media-streaming platforms are abundant in content, and the *de facto* go-to online encyclopedia, English Wikipedia, lists more than seven million content pages as of July 2025. Various methods have been explored in the field of information retrieval to improve the navigability of large collections and ultimately present users with relevant results. Query reformulation constitutes a general framework for improving the "coordination match" (Krovetz and Croft, 1992) to increase retrieval performance by adding, reweighting, or removing search query terms. The addition of synonymous or semantically similar terms to the initial query is known as *query expansion*, which allows the search engine to potentially find more relevant documents that match these terms.

However, queries may suffer from ambiguity when the surface forms of the terms have different meanings. This is especially apparent for short queries, where a search for "python" may return documents related to the programming language, family of snakes, or any other meaning referred to by this surface form. While query expansion may still help shed light on the fact that all these different senses exist, it will likely take several iterations for the user to arrive at the results they expected, so much so that most users will pre-empt this process by performing manual query expansion at the first iteration, providing a query such as "python programming language" to narrow down the set of relevant documents. However, this requires knowledge that the query is ambiguous in the first place.

Personalised query expansion (PQE) can largely avoid this manual query modification by selecting expansion terms based on user context. Broadly, this context is any side-channel information that charac-

[a] https://orcid.org/0000-0003-4919-9033
[b] https://orcid.org/0000-0002-2843-3137

terises a user's potential information needs, such as query histories, page views, or social media content. This approach inherently requires a language model for both the "user language" and the corpus.

The dominant approach to language modelling is *embedding*, by which words are represented as dense high-dimensional vectors. Recently, the contextualised word embeddings underlying large language models (LLMs) have advanced information retrieval; however, their size and computational cost during training and inference can become an obstacle for constrained settings. In contrast, static embedding models, such as word2vec and GloVe, remain efficient to train and query, even on consumer-grade hardware. Furthermore, they facilitate the development of explainable intelligent systems, as querying the embedding for similar terms directly relates to the training set, specifically a user profile, rather than extensive Web-scale collections, such as the Common Crawl.

In this study, we propose and analyse a refinement of PQE, which we refer to as *Personalised Semantic Query Expansion* (PSemQE). PSemQE expands queries with semantically close terms with respect to the user's context by performing a nearest-neighbour search in the embedding space. Crucially, it uses local embeddings of user profiles only during query expansion and defers actual retrieval to a downstream search engine.

Although similar systems have been explored previously, their ability to disambiguate queries for the user's benefit is not well understood. This study provides insights into the extent to which PSemQE can perform word-sense disambiguation in a user-centric manner.

Our contributions are summarised as follows:

- Detailed analysis of the rank dynamics governing personalised query expansion for short, ambiguous queries;
- Outlier analysis of cases where expansion produces inferior results;
- Collection of a labelled dataset for evaluating query disambiguation, as well as synthetic user modelling.

The remainder of this paper is structured as follows: Section 2 discusses related work and positions this study in a broader context. In Section 3, we provide an operationalisation and describe our approach to data collection and analysis, as well as the experimental design. The results are reported and discussed in Section 4. Finally, Section 5 concludes the paper and provides an outlook on future research.

2 RELATED WORK

In the following section, we review the literature related to our study. We begin by reviewing query expansion in general before moving to the subfield of personalised query expansion. As language modelling is an important part of any query modification system, we conclude this section by reviewing the key contributions in this field, including recent approaches using large language models.

Query Expansion and Ambiguity

Query expansion is a type of query reformulation that extends the initial query with additional terms to improve retrieval performance. Suitable candidates for expansion terms can be sourced from the search results themselves or from "knowledge structures" (Efthimiadis, 1996).

Relying on search results as a source for expansion terms constitutes a form of relevance feedback (Manning et al., 2008), where the results retrieved from an initial search using the original query are presented to the user, who judges which documents are relevant to their information needs. To automate this procedure, the top k results of an initial search may be regarded as relevant; this is known as *pseudo-relevance feedback*.

However, these initial search results may contain irrelevant documents or miss relevant documents owing to the ambiguity of words in the corpus and/or query (Krovetz and Croft, 1992). Therefore, imbalances in the corpus may be reflected in the initial retrieval if an ambiguous term is overrepresented, leading to query drift (Croft et al., 2001). *Clarity* has been suggested as a measure by which users can be guided in disambiguating queries (Croft et al., 2001; Cronen-Townsend and Croft, 2002).

Personalised QE

The prevalence of user-generated data on the Web has been influential in the development of personalised query expansion. By considering users' actions outside the search context, these approaches attempt to produce results that are closely tied to users' interests.

A wide variety of sources from which expansion terms can be drawn have been explored, such as emails (Chirita et al., 2007), social tagging (Bouadjenek et al., 2019), and the fusion of social annotations with knowledge bases (Zhou et al., 2017).

For instance, Ould Amer et al. (Ould-Amer et al., 2016) investigated personalised book search by constructing word2vec embeddings of users' item cata-

logues. While they found that the personalised setting outperformed the non-personalised setting only when a small number of expansion terms were included, they explained this finding with the quality of the corpus rather than the method in general.

Recently, Bassani et al. (Bassani et al., 2023) used contextual word embeddings for personalised QE by aligning topical clusters of user and document embeddings and selecting the most appropriate terms using a nearest-neighbour search.

Language modelling

Following Efthimiadis' classification, expansion terms can also be sourced from knowledge structures that are not directly tied to the search process, such as domain-specific or general thesauri. Language models, such as word embeddings, can also be considered a form of knowledge structure.

Popular approaches for training word embeddings include *word2vec* (Mikolov et al., 2013a; Mikolov et al., 2013b) and *GloVe* (Pennington et al., 2014). These vectors are derived such that words with similar meanings correspond to vectors that are also close in the embedding space; cosine distance is a common choice as a metric. This property also makes embeddings an interesting resource for candidate terms for query expansion because words that are similar to the original query can be chosen for expansion. Roy et al. (Roy et al., 2016) studied different methods to extract candidate terms from word2vec embeddings; among others, they explored the compositionality of word vectors to extend a query with terms similar to a bigram decomposition of the initial query.

Large language models, such as BERT (Devlin et al., 2019), use contextualised embeddings, where individual words are represented by multiple vectors depending on their context. Contextualised word embeddings have been shown to perform well in capturing senses and performing word sense disambiguation (Wiedemann et al., 2019; Loureiro et al., 2021). Specifically, SensEmBert (Scarlini et al., 2020) fuses BERT with semantic networks to produce multilingual sense embeddings.

Naseri et al. (Naseri et al., 2021) used contextualised embeddings provided by BERT for query expansion, reporting improvements in average precision on TREC test collections compared to static models. Wang et al. (Wang et al., 2023) and Jagerman et al. (Jagerman et al., 2023) used LLMs to generate pseudo-documents and expansion terms by utilizing the inference capabilities of these models. The model output was subsequently appended to the original query, with improvements in retrieval performance across several test collections.

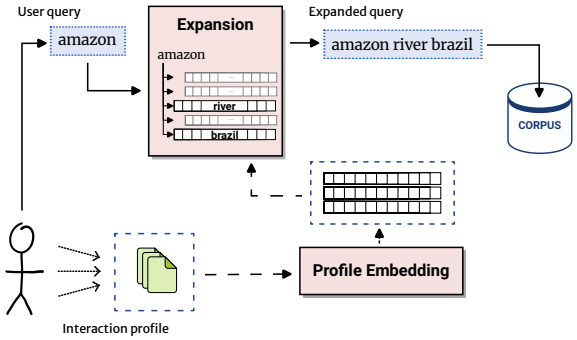

Figure 1: Components of the expansion system. Users' past interactions are recorded and represented via word embeddings. Any query issued by the user passes through the expansion component, where the top-k similar terms are appended.

While large models have been shown to perform well in word sense disambiguation, this remains an open question for static models. However, these smaller models are efficient even in constrained settings and constitute a "small world" model of a user's language, thus aiding explainability for downstream tasks. To the best of our knowledge, this is the first study to investigate the "disambiguation power" of personalised query expansion using local embeddings.

3 METHODOLOGY

The query expansion system we consider comprises three key components: In an offline procedure, word embeddings are trained on individual user profiles to capture semantic relationships between terms within the user's context. The expansion component uses these embeddings to extend the original query with semantically similar expansion terms. Finally, the search engine processes the query and retrieves and ranks the documents. The components and their interactions are shown in Figure 1.

This architecture enables personalised semantic query expansion by leveraging user profile information to disambiguate short, ambiguous queries and retrieve more relevant results tailored to the user's interests and context.

To better understand the extent to which PSemQE can disambiguate queries, we addressed the following research questions:

RQ1 For short, ambiguous queries, can PSemQE retrieve documents that the base query can't?

RQ2 Are relevant documents ranked higher under PSemQE than the base query?

RQ3 Does PSemQE yield more relevant documents than the base query?

In the following section, we operationalise the key concepts for our work and then document our data collection process. In particular, we describe how we built synthetic user profiles in the absence of ground truth.

3.1 Operationalization

User context

We broadly define user context as any data that characterises the interests of a user. For this study, we used browsing histories containing items that a user had viewed previously. We assume that these histories implicitly describe a user's interests and understanding of the domain.

Query disambiguation

We consider a query ambiguous if it consists of one or more words that exhibit multiple senses. Query disambiguation is the process of inferring the sense intended by the user and having the search results reflect this sense.

We chose to model the problem setting on the English Wikipedia owing to the availability of ambiguous lemmata along with possible disambiguations. The titles of dedicated *disambiguation pages* refer to ambiguous lemmata, and the body contains a list of links to pages that disambiguate the term. For the evaluation data, we selected the title of a disambiguation page as the query and selectively sampled two pages from the list as the disambiguation targets. Given a user profile that consists of articles relating to the sense of one of the targets, we regard disambiguation as successful if the target is present in the search results. An illustrative example: for the query "Amazon" and a user showing interest in geography, ecosystems, and climate, the query is successfully disambiguated if the results contain the target article "Amazon rainforest".

Synthetic user profiles

While the raw Wikipedia Web request stream (Wikipedia contributors, 2025a) contains the requesting IP address and requested URL, this data is considered sensitive personal information according to the Wikimedia Foundation's privacy policy (Wikipedia contributors, 2025d) and is not available to the general public. Anonymised aggregated data are available in the form of pageview dumps (Wikipedia contributors, 2025b) or the Wikipedia Clickstream dataset (Wikipedia contributors, 2025c), but these can only provide an approximation of real users' browsing habits. While Arora et al. (Arora et al., 2022) argued that in certain cases, "synthetic data is enough", they also noted that real user traces are useful for tracking the patterns of individual users. We reproduced their approach in preliminary experiments but found the profiles generated in this way to be too broad in terms of the domains of interest of the resulting profiles, and no meaningful keywords were extracted from these profiles. These preliminary findings suggest that to perform meaningful personalised query expansion, user profiles should revolve around well-defined topics.

Owing to the lack of publicly available user profiles for Wikipedia, we generated synthetic profiles based on a simple knowledge graph, the process of which is described in Section 3.2. The synthetic profiles we collected consist of an unordered set of unique page IDs of Wikipedia articles and constitute a model of users with specific interests browsing Wikipedia.

3.2 Data collection

As a consequence of the operationalisation given in the preceding section, we require two datasets for our evaluation: ambiguous Wikipedia articles, along with possible disambiguations, and user profiles. This section describes the collection strategies for both datasets.

Ambiguous articles

We selectively sampled a set of 14 disambiguation pages from the category listing for all article disambiguation pages[1]; we refer to these as the *parent term* or simply *parent* in the remainder of this paper. From each disambiguation page, we again selectively sampled two pages as disambiguation targets, from which we extracted the full-text content of the latest revision[2]. The raw contents were stripped from wikimarkup[3] in preparation for further processing. We collected 28 $(parent, target)$-tuples, two of which are listed in Table 1.

User profiles

As outlined in Section 3.1, for this study, we regard a user profile as an unordered set of Wikipedia articles

[1] https://en.wikipedia.org/wiki/Category:All_article_disambiguation_pages

[2] https://www.mediawiki.org/wiki/API:Main_page

[3] https://github.com/earwig/mwparserfromhell

Table 1: Two example parent terms and the corresponding target articles to gauge disambiguation; each target is a concrete article linked to from the parent's disambiguation page.

Parent ID	Parent title	Target ID	Target title
29621629	Amazon	1701 90451	Amazon River Amazon (company)
8239	Dylan	4637590 8741	Bob Dylan Dylan (programming language)

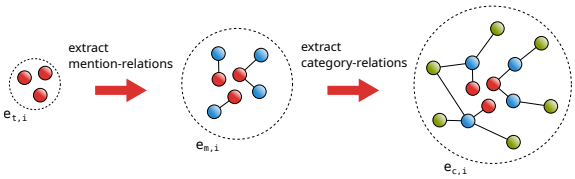

Figure 2: Construction of the knowledge graph

with the requirement that the articles share a common theme. In this section, we describe the process of generating synthetic profiles using random walks on a knowledge graph.

Knowledge graphs (KGs) model facts regarding a domain using a graph structure. In KGs, nodes are typically referred to as *entities* E, and the edges between them as *relations* R; formally, $KG = (E,R)$. The entities in our knowledge graph KG_{wiki} consist of Wikipedia articles, which we connect through two types of undirected relations: r_m, a *mentions*-relation indicating that one article is mentioned in the body of the other, and r_c, a *category*-relation indicating that the endpoints have a common Wikipedia category. Figure 2 illustrates the iterative procedure used to generate the knowledge graph.

Initially, KG_{wiki} contained the 28 target articles as entities (cf. Section 3.2); we refer to these as e_t. In the first step, we identified articles mentioned in the body of the target articles, added them to the graph as new entities e_m, and connected each to the target in which they were mentioned through a new relation r_m. In the second step, for each e_m, we determined which Wikipedia category they belonged to and collected further articles from these categories. These are again added to the graph as entities e_c, connected to e_m through a *category*-relation.

To identify *mentions*-relations, we used DBpedia Spotlight(Daiber et al., 2013), a service for annotating text with Wikipedia articles that are indexed by DBpedia. Spotlight identifies surface terms in the text that refer to DBpedia entities in a context-aware way. For instance, the surface form "boroughs" in the Wikipedia article on Berlin is linked to the concept "Boroughs and neighborhoods of Berlin", We queried the Spotlight API using the default settings, notably a confidence setting of 0.5. For each identified DBpedia entity, we fetched the Wikipedia page ID and retrieved the full text of the latest revision.

For each article obtained in this way, we extracted the category labels from the wiki-markup and queried the Wikipedia API for up to ten members of each category. As before, we retrieved the full text of the latest revision of each article.

This process iteratively builds a knowledge graph of articles, starting with the target articles. By including articles mentioned in the text as well as those related through mutual categories, we intend to cover strongly related concepts alongside broader associations via categories.

The final user profiles are then generated as random walks on KG_{wiki}. We generated one profile per target article, and each walk originated from the node corresponding to the target article, terminating after a maximum of 40 steps. Because KG_{wiki} is unweighted, each transition occurs with probability $\frac{1}{d}$, where d denotes the degree of the current node. For each node visited during the walk, we recorded the Wikipedia page ID of the corresponding article. The ID of the target article was dropped from the sequence and recorded as an identifier to retrieve each profile in the experiment. The profile vectors obtained in this way consist of Wikipedia articles that share a common theme: one article may mention another, or a related concept that is subsequently researched, or the articles may share a common category. These profiles approximate users browsing a knowledge base, whose journeys eventually lead to the target article.

Finally, we trained a word-embedding model for each profile vector using the full text of each article. As before, we retrieved the latest revision, removed markup, and performed minimal pre-processing using *gensim* (Řehůřek and Sojka, 2010): tokenisation on whitespace, lower-case normalisation, removal of punctuation characters, and pruning of tokens shorter than two and longer than 15 characters; stop words were removed using a stop word corpus (Bird et al., 2009).

The collection of documents comprising a single profile was then used to train the word2vec model. We used the default settings provided by *gensim*, with

the following deviations: the dimension of the embedding vectors was set to 300, using a window size of three and a minimum word frequency of one; models were trained over seven epochs.

3.3 Experimental design

To evaluate whether user profiles can be used to disambiguate search queries, we devised an experimental setting that simulates users searching a knowledge base. Predefined ambiguous search queries corresponding to the titles of Wikipedia disambiguation pages were submitted to a search engine. The queries were provided in two forms: the base case with no further keywords appended and the expansion case with additional keywords appended to the base form. Expansion keywords were retrieved from the word embeddings that constitute user profiles. For each word in the query, the two closest neighbours in terms of the cosine distance of their embedding vectors were determined. Out-of-vocabulary misses were ignored, and if none of the query words appeared in the embedding, the expanded and plain queries were the same. Let q_{orig} denote the original query, $\langle w_1, \ldots, w_n \rangle$ the set of expansion terms, and $\oplus$ the string concatenation operator. The two queries are then given by

$$q_{base} = q_{orig}$$
$$q_{expansion} = q_{orig} \oplus \langle w_1, \ldots, w_n \rangle$$

We used this setup for two distinct experiments that differed in the size and composition of the search corpora but otherwise followed the same protocol. Recall from Section 3.2 that we collected 14 ambiguous parent terms, and two articles relate to each parent that disambiguate it, leading to a total of 28 $(parent, target)$ tuples. In this collection, each parent appears twice, with one of the two associated articles serving as the target. For each tuple, the two queries were constructed as given above, using the *parent*-term as q_{orig} and the profile built around the respective target article as the source of expansion terms. The goal is to search the corpus for the target using q_{base} and $q_{expansion}$. An example best illustrates this approach: for the parent term "Amazon", the first target instance is "Amazon River"; q_{base} is set to "amazon", and based on the profile around this target, $q_{expansion}$ is determined as "amazon river brazil". These two queries were submitted to the search engine, and the results were analysed with respect to the target instance. This process is repeated for the second target instance, "Amazon (company)", and so forth, for each of the remaining tuples.

In the first experiment, referred to as SMALL in the remainder of this paper, the intention was to evaluate disambiguation on corpora containing highly similar documents. These corpora contain a small number of documents related to one ambiguous query. They are generated dynamically, once for each parent term, and always contain both instances of the target article, plus up to five other articles sampled from the same disambiguation page, which serve as "flares". Table 2 illustrates the corpus configurations for the experiments. This leads to 14 distinct corpora used in SMALL, each consisting of up to seven documents:

$$(target_a, target_b, flare_1, ..., flare_5)$$

Table 2: Setup for experiments SMALL and LARGE.

Experiment	#Documents	#Corpora
SMALL	$[2,7]$	14
LARGE	3576	1

For the second experiment (LARGE), we increased the corpus size and included all articles collected during the construction of the knowledge graph. The idea is to mirror a real-world corpus that comprises a multitude of different items. The corpus is compiled once and reused for all subsequent queries.

The experiments are conducted using *lunr.py*[4], which uses BM25 internally to weight results. The search results were retrieved in the order determined by *lunr* and were limited to the top 100 results. In total, we collected 56 result sets (14 parents $\times$ 2 targets $\times$ 2 query types).

3.4 Evaluation

We observed that four queries were not expanded, resulting in q_{base} and $q_{expansion}$ being identical. In each of these instances, the absence of expansion terms can be attributed to an out-of-vocabulary miss in one or both of the word embedding models utilised during the expansion phase. As we cannot reason about differences in retrieval for identical base and expansion queries, we omit these cases from the evaluation, reducing the overall number of result sets from 56 to 40 by 16 (4 parents $\times$ 2 targets $\times$ 2 query types). This resulted in 20 sets of results per experiment for evaluation.

After filtering out unexpanded trials, the result sets for SMALL contain a mean of 5.8 ($\pm$1.11) results for the base case and 5.85 ($\pm$1.09) for the expansion case. The result sets for LARGE contained 89.2 items on average ($\pm$20.3) for the base group, and 99 items ($\pm$4.47) for the expansion group. Figure 3 shows the distribution of the set size for both experiments.

[4] https://github.com/yeraydiazdiaz/lunr.py

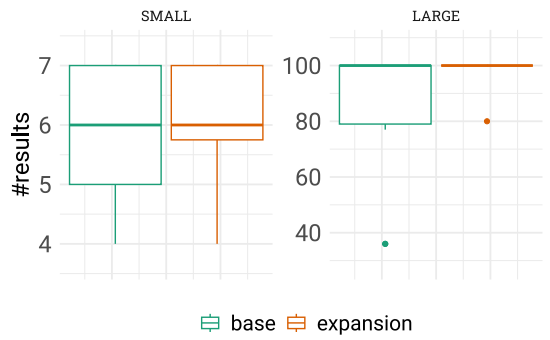

Figure 3: Distribution of the result set size across the two experiments, SMALL and LARGE.

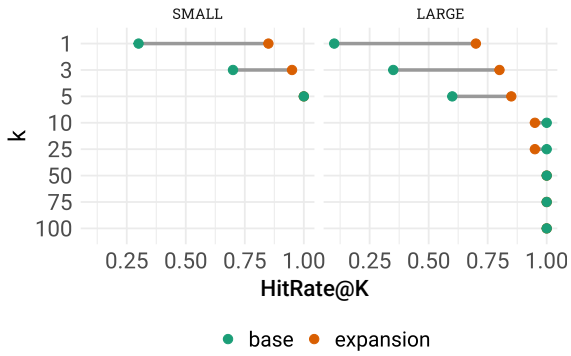

Figure 4: HitRate@K over all queries for selected K. Owing to the size of the SMALL corpus, no query returned more than seven results.

We measured whether the target article was part of the result list on a binary level, as well as the numeric rank of the target in the list. In addition, we measured the overlap of the sets of categories appearing in the results with those appearing in the profile.

4 RESULTS & DISCUSSION

4.1 Results

The following section documents our results, which are discussed in Section 4.2. We report our findings on hit rate, rank distribution, and category overlap. This section concludes with an analysis of observed outliers.

4.1.1 Hit rate

Recall that we use target articles as proxies for disambiguation: they cover two of the multiple possible senses of the parent term. Therefore, if a search can retrieve the predetermined target, we assume that the query has been disambiguated. However, if the target was not retrieved, the intended meaning was not inferred correctly. *HitRate@K* is the fraction of queries that contained at least one relevant document over all queries:

$$HitRate@K = \frac{Hit@K}{|Q|} \tag{1}$$

Here, $Hit@K$ is a binary indicator determining presence or absence of the target article t in the set of results R_q for a query q at the cutoff K:

$$Hit@K = \begin{cases} 1 & \text{if } t \in R_q \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Note that each query $q \in Q$ returns results, as the search corpus always contains at least two articles (the targets) that match the query (cf. Section 3.3). In other words, R_q in Eq. 2 is never empty.

Figure 4 shows the $HitRate@K$ for fixed values of $K \in \{1,3,5,10,25,50,75,100\}$. Across both experiments, we found that for lower K and hence shorter result lists, query expansion outperformed the base case in terms of $HitRate$ and retrieved the target article in more cases. As the length of the result set increases, the rate approaches 1.0 for both *base* and *expansion*. In other words, the longer the result list, the more likely it is to include the target. This is unsurprising and similar to the observation that perfect *recall* can be achieved by simply returning all documents in the collection (cf. (Manning et al., 2008)).

It is important to consider the two data points in LARGE at $k = 10$ and $k = 25$. In these cases, q_{base} achieves a higher $HitRate@K$ than $q_{expansion}$; that is, using only the plain query, the target is present in more result sets. In other words, outliers exist for which query expansion fails to retrieve the target. Upon closer examination, we discovered that this issue affected only a single query, and that the observation stemmed from the specific expansion terms employed. A more detailed analysis of this observation is presented in Section 4.1.4.

4.1.2 Distribution of ranks

Next, we evaluated the rank distribution of the target articles. Previous work indicates that users tend to regard the first ten search results highly (Silverstein et al., 1999; Jansen et al., 2000). Thus, a document's rank is an important factor in the subjective assessment of relevance, and the lower the rank, the higher the chance that the user will direct their interest to the document.

Figure 5 shows the distribution of ranks for both experiments across the two groups. The expansion group tends to yield the target article at lower ranks

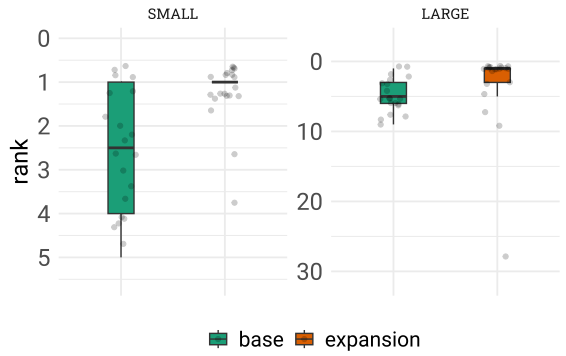

Figure 5: Distribution of ranks; lower rank is better

Table 3: Summary statistics of rank by query type, pooled over the experiments. A Mann-Whitney U test of *base* producing larger ranks than *exp.* was significant at $p < 0.001$.

type	Mean (SD)	Median (IQR)	Range
base	3.7 (2.2)	3.0 (2.0, 5.0)	1.0, 9.0
exp.	2.4 (4.5)	1.0 (1.0, 1.0)	1.0, 28.0

than the plain query.

Table 3 shows the characteristics of the distribution of target ranks across the two folds, *plain* and *expansion*. We see that the median rank improves from plain to expansion by two ranks, from 3.0 to 1.0. We conduct a one-tailed Mann-Whitney-U test ($\alpha = 0.001$) with a null hypothesis defined as *plain* producing ranks lower than or equal to *expansion*. As indicated in the table, we reject the null ($p = 2.771 \times 10^{-6}$) and accept the alternative hypothesis that *plain* produces greater ranks than *expansion*. Thus, we conclude that query expansion tends to produce lower ranks for target articles than plain queries, placing them higher on the list of results.

4.1.3 Category overlap

Our final analysis investigated the overall relevance of the results, specifically whether PSemQE would return more relevant results than the base query. We gauge relevance via Wikipedia categories: for all articles in the search corpus, we collect the assigned category tags. Then, we determine the proportion of these categories that are also present in the search results As described in Section 4.1.1, we determined this overlap for different subsets of the results. Let C_{user} denote the set of categories appearing in a user profile and C^k_{search} the set of categories contained in the top-k results. Then, we define $overlap@k$ as

$$overlap@k = \frac{|C_{user} \cap C^k_{search}|}{|C^k_{search}|}$$

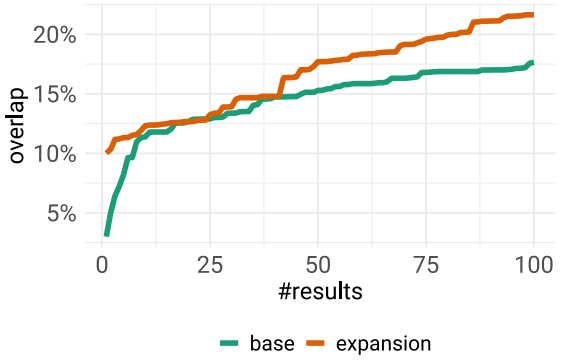

Figure 6: Overlap between search result categories and user profile

The mean $overlap@k$ is shown in Figure 6. As shown in the figure, query expansion consistently achieves a higher overlap between the two category sets than the plain variant. This is true not only for the experiment on the SMALL corpus, where the degree of overlap tops off at 11% owing to the limited size of the search corpus, but also for the LARGE corpus, where query expansion starts off with a higher overlap even for a single result ($k = 1$) than a plain query. In this case, as we increased the size of the result set ($k \geq 100$), the degree of overlap approached 25%. Thus, query expansion can better reproduce users' interests than a plain query, even for comparably small result lists of 50 items or fewer.

4.1.4 Outlier analysis

In this section, we revisit the outliers observed during the analysis in Section 4. Evaluating $HitRate@K$, we observed that on the LARGE corpus with query expansion, some targets were not retrieved, even for result lists of length 25 (cf. Figure 4). This observation is tightly coupled with the outlier in Figure 5, where one article is ranked at position 28 with query expansion. The latter observation explains the former: because of its rank, the article only appears in the result lists with a length of 28 or greater.

The article in question is "Amazon (company)", and the expanded query presented to the search engine was "amazon products linux". In the respective embedding, the terms "products" ($\cos(\theta) = 0.99$) and "linux" ($\cos(\theta) = 0.96$) were closest to "amazon" based on their vectors' cosine similarity. Overall, the embedding captured the notion of "amazon" in a technology context, with terms such as *developers*, *computer*, *app*, and *startup* among the 20 most similar terms. However, the term *linux* appears only in one of the 31 articles used to train the embedding: "Android (operating system)". Notably, the target article "Amazon (company)" also does not contain the term

linux.

As mentioned in Section 3.3, the documents were scored using the BM25 model, which is sensitive to term frequencies. For document D and query $Q = Q_0, \ldots, q_n$, the document-level score $BM25(Q,D)$ is (cf. (Manning et al., 2008))

$$\sum_{i=0}^{n} IDF(q_i) \cdot \frac{tf(q_i,D) \cdot (k_1+1)}{tf(q_i,D) + k_1 \cdot (1 - b + b\frac{DL}{AVDL})}$$

Here, k_1 and b are tunable parameters (discussed below), $tf(q_i,D)$ is the raw term frequency of query word q_i in document D, $IDF(q_i)$ is the inverse document frequency of q_i in the corpus, and DL and $AVDL$ are the current document's length and average document length across the entire corpus, respectively. The final score for document D and query $Q = q_0, \ldots, q_n$ is then computed as follows (cf. (Manning et al., 2008)):

In Table 4, we compare the raw term counts and the BM25 score for the outlier with the top three results in the set. Although the target article achieves marginally higher BM25 scores for both *product* and *linux*, these differences do not outweigh the absence of *linux*: with a term frequency of zero, the summand for this query word also becomes zero.

To further gauge the impact of k_1, b, and $tf(\cdot)$ on the rank of "Amazon (company)", we conducted a sensitivity analysis for each parameter. k_1 determines the scaling of term frequencies, with suggested values in [1.2, 2] (Spärck Jones et al., 1998). At the extremes, this parameter turns BM25 into a binary model ($k_1 = 0$), detecting only term presence, or scales the score linearly in $tf(\cdot)$ (large k_1) as if using raw term frequency (Manning et al., 2008). The parameter b influences the score's dependence on document length and ranges from $[0,1]$; for $b = 0$, document length is not factored in, $b = 1$ fully scales the score by document length.

Figure 7 shows the results of the parameter analysis. We vary k_1 over $[0,20]$ and b over $[0,1]$, as well as the raw term frequency only for the target article, over the range $[0,41]$, up to the maximum term frequency in the set of these four articles[5]. To achieve the highest score for the target article, we would have to set $k_1 = 7.5$, which is well above the recommended range. Changing the document length scaling factor b does not affect the target's rank. However, if the term *linux* appeared at least twice in the document, it would rank among the top three results; if it appeared at least four times, it would take the top position.

[5]Varying the term frequency also influences the term's IDF; as the difference is only on the order of 1×10^{-2}, we omit further analysis of this effect.

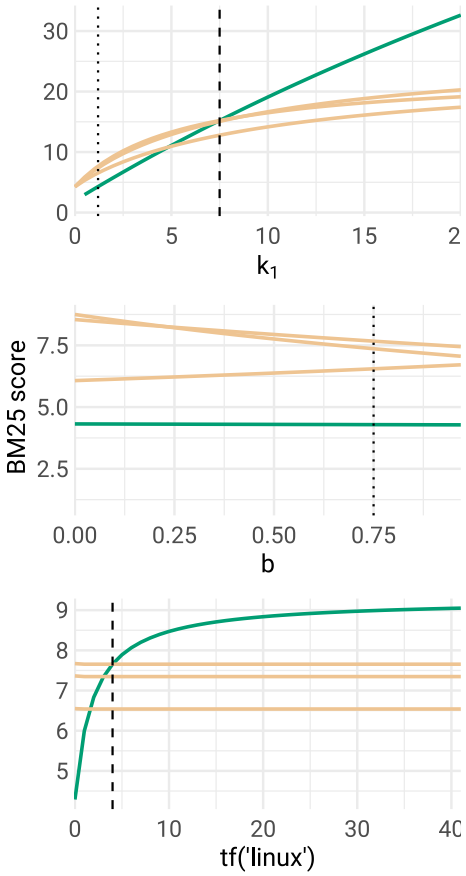

Figure 7: BM25 score analysis for k_1, b, and term counts of *linux*. The line corresponding to the target-article is highlighted; dashed lines indicate the threshold at which the target-article score exceeds the others', dotted lines indicate the values of k_1 and b set for the experiments

An interesting observation is that although the target article was ranked far down the list using the expanded query, the search results nevertheless contained more relevant results in terms of category overlap than the base query, especially for ranks 1 through 3 and up to 10, as illustrated in Figure 8.

In summary, we found that the target article was not retrieved among the top results because of the expansion with a keyword that was not relevant to this document. Altering the tunable parameter k_1 of the BM25 model to scale the remaining query-term frequencies resolves this issue, although the consequences for other queries are unclear. A more viable approach would be to not augment the query with irrelevant keywords, although "irrelevant" is subjective in this setting: we only know that the keyword is irrelevant because we know the target beforehand. A better approach would be to further refine the models used for the profile embeddings by, for example,

Table 4: Raw term counts and BM25 scores for the outlier article, compared to the top three results

		term counts			BM25			
rank	title	*product*	*amazon*	*linux*	*product*	*amazon*	*linux*	Σ
1	Tablet computer	15	11	13	0.87	2.67	4.13	7.67
2	Android (operating system)	9	10	41	0.66	2.24	4.47	7.36
3	Audible (service)	8	8	1	0.92	3.00	2.63	6.55
28	Amazon (company)	48	167	0	0.98	3.30	0.00	4.29

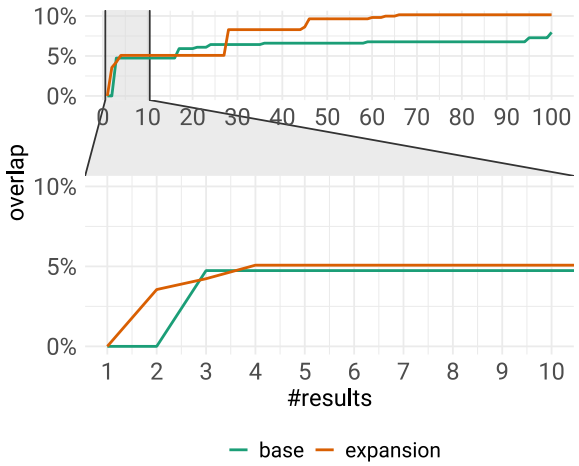

Figure 8: Category overlap in the search for "Amazon (company)".

pruning words from the vocabulary that occur below a given threshold or shrinking the window size of the word2vec-algorithm to avoid loose associations between terms.

4.2 Discussion

In **RQ1**, we asked whether query expansion could be used to disambiguate queries exhibiting multiple senses and return results that the base query would not reveal. To answer this question, we measured successful disambiguation by retrieving a predefined target article. We found that both the plain query and the expanded version returned the target article at least within the first ten results. Therefore, both approaches yield the intended sense at some position in the result list, and PSemQE only outperforms on top-k lists for small k.

RQ2 then asked whether query expansion could rank relevant documents higher. Our analysis of ranks showed that personalised semantic query expansion has a significant tendency to produce lower ranks, that is, it places the article higher in the list. Thus, personalised semantic query expansion is a supportive measure for inferring a user's intended sense based on their prior interactions with the system.

Finally, in **RQ3**, we asked whether PSemQE could produce results that are more relevant to a user than a simple query. We measured relevance via categories assigned to articles in terms of the proportion of categories from a user profile that were present in the search results. We observed that query expansion can result in more of the user's categories being present in the result list, especially in the long tail of 50 results or more. This may indicate that as users browse through more items through pagination, lazy loading, or other measures, the relevance of the results does not necessarily degrade.

In summary, we conclude that personalised query expansion can be a supportive aid in disambiguating queries to the intended sense and, overall, can return more potentially relevant documents than a plain query would.

4.3 Limitations

The approach we explored has two limitations: First, we rely on synthetic user profiles that we generated to model a narrowly focused information need; real user profiles may not be as strictly focused on a single domain. However, as long as there is a dominant context that allows the framing of ambiguous terms, the profile should be suitable for this approach. We believe this to be true, especially for libraries, archives, and research databases, which are often consulted by users with well-defined research interests.

Second, during profile generation, we controlled the size of the local profile corpus. For users whose profiles are sparse or, in the case of new users, entirely empty, no meaningful word embeddings are available. This sensitivity to cold start is inherent in any personalisation facet of information retrieval, such as recommender systems. In these cases, it may be desirable to focus on interactive query disambiguation, detecting ambiguity via, for example, thesauri or measures such as clarity (Croft et al., 2001; Cronen-Townsend and Croft, 2002). Additionally, users with sparse profiles may be prompted to provide seed documents that have been relevant to them in the past, in the form of short excerpts or abstracts.

5 CONCLUSION

Query expansion is an important component of information retrieval (IR) systems. It allows users to bridge the query-document vocabulary gap without having to invest mental effort in framing their information needs in the context of the queried collection.

This study demonstrated the potential of Personalised Semantic Query Expansion (PSemQE) as an effective approach for disambiguating short, ambiguous queries in IR systems. The experimental results showed that PSemQE could successfully retrieve relevant documents that matched the intended sense of ambiguous terms, often ranking them higher than a base query without expansion. Furthermore, PSemQE consistently produced result sets with higher relevance to user interests, as measured by the category overlap between search results and user profiles.

Although PSemQE shows promise in improving search relevance and user experience, some limitations must be considered. Reliance on synthetic user profiles focused on narrow domains may not fully represent real-world user behaviour. Additionally, this approach may face challenges with new or inactive users owing to insufficient profile data for meaningful word embeddings.

Further research is required to explore ways to address these limitations, such as investigating methods for building more diverse synthetic profiles when no real-world data are available, developing strategies for cold-start scenarios, and incorporating interactive disambiguation techniques. Overall, this study shows that systems for personalised query expansion can act as disambiguation engines when they integrate user information and reliably return meaningful responses for short, ambiguous queries.

DATA AVAILABILITY

The code to reproduce our experiments is published at https://github.com/baumanno/psqe-2024.
Data is published via Zenodo at https://doi.org/10.5281/zenodo.10729730.

ACKNOWLEDGMENTS

This article is the outcome of research conducted within the Africa Multiple Cluster of Excellence at the University of Bayreuth, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2052/1 – 390713894.

REFERENCES

Arora, A., Gerlach, M., Piccardi, T., García-Durán, A., and West, R. (2022). Wikipedia reader navigation: When synthetic data is enough. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 16–26, New York, NY, USA. ACM.

Bassani, E., Tonellotto, N., and Pasi, G. (2023). Personalized Query Expansion with Contextual Word Embeddings. *ACM Trans. Inf. Syst.*, 42(2):61:1–61:35.

Bird, S., Loper, E., and Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly.

Bouadjenek, M. R., Hacid, H., and Bouzeghoub, M. (2019). Personalized social query expansion using social annotations. In Hameurlain, A., Wagner, R., Morvan, F., and Tamine, L., editors, *Transactions on Large-Scale Data- and Knowledge-Centered Systems XL*, pages 1–25, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chirita, P. A., Firan, C. S., and Nejdl, W. (2007). Personalized query expansion for the web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 7–14, New York, NY, USA. Association for Computing Machinery.

Croft, W. B., Cronen-Townsend, S., and Lavrenko, V. (2001). Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshops / Conferences*.

Cronen-Townsend, S. and Croft, W. B. (2002). Quantifying query ambiguity. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 104–109, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, page 121–124, New York, NY, USA. ACM.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Efthimiadis, E. N. (1996). Query expansion. In Williams, M. E., editor, *Annual Review of Information Systems and Technology (ARIST)*, volume 31, pages 121–187.

Jagerman, R., Zhuang, H., Qin, Z., Wang, X., and Bendersky, M. (2023). Query expansion by prompting large

language models. In *Gen-IR@SIGIR 2023: The First Workshop on Generative Information Retrieval*.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207–227.

Krovetz, R. and Croft, W. B. (1992). Lexical ambiguity and information retrieval. *ACM Trans. Inf. Syst.*, 10(2):115–141.

Loureiro, D., Rezaee, K., Pilehvar, M. T., and Camacho-Collados, J. (2021). Analysis and evaluation of language models for word sense disambiguation. *International Conference on Computational Logic*.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013*, Scottsdale, Arizona, USA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Naseri, S., Dalton, J., Yates, A., and Allan, J. (2021). CEQE: Contextualized Embeddings for Query Expansion. In Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., and Sebastiani, F., editors, *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 467–482, Cham. Springer International Publishing.

Ould-Amer, N., Mulhem, P., and Géry, M. (2016). Toward Word Embedding for Personalized Information Retrieval. In *Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval*.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Roy, D., Paul, D., Mitra, M., and Garain, U. (2016). Using Word Embeddings for Automatic Query Expansion. In *Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval*.

Scarlini, B., Pasini, T., and Navigli, R. (2020). Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. *AAAI Conference on Artificial Intelligence*.

Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large web search engine query log. *ACM SIGIR Forum*, 33(1):6–12.

Spärck Jones, K., Walker, S., and Robertson, S. (1998). A probabilistic model of information and retrieval: Development and status. Technical Report UCAM-CL-TR-446, University of Cambridge, Computer Laboratory.

Wang, L., Yang, N., and Wei, F. (2023). Query2doc: Query Expansion with Large Language Models. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Wiedemann, G., Remus, S., Chawla, A., and Biemann, C. (2019). Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Wikipedia contributors (2025a). Data Platform/Data Lake/Traffic/Webrequest — Wikitech. [Online; accessed 25th January 2025].

Wikipedia contributors (2025b). Research:Page view — Wikimedia Meta-Wiki. [Online; accessed 25th January 2025].

Wikipedia contributors (2025c). Research:Wikipedia Clickstream — Wikimedia Meta-Wiki. [Online; accessed 25th January 2025].

Wikipedia contributors (2025d). Wikimedia Foundation Privacy Policy — Wikimedia Foundation Governance Wiki. [Online; accessed 25th January 2025].

Zhou, D., Wu, X., Zhao, W., Lawless, S., and Liu, J. (2017). Query expansion with enriched user profiles for personalized search utilizing folksonomy data. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1536–1548.