Bridging the Terminology Gap in Web Archive Search*

Klaus Berberich Srikanta Bedathur Mauro Sozio Gerhard Weikum

Max-Planck Institute for Informatics
Saarbrücken, Germany
{kberberi, bedathur, msozio, weikum}@mpi-inf.mpg.de

ABSTRACT

Web archives play an important role in preserving our cultural heritage for future generations. When searching them, a serious problem arises from the fact that terminology evolves constantly. Since today's users formulate queries using current terminology, old but relevant documents are often not retrieved. The query saint petersburg museum, for instance, does not retrieve documents from the 1970s about museums in Leningrad (the former name of Saint Petersburg).

We address this problem by determining query reformulations that paraphrase the user's information need using terminology prevalent in the past. A measure of across-time semantic similarity that assesses the degree of relatedness between two terms when used at different times is proposed. Using this measure as a crucial building block, we propose a novel query reformulation technique based on a hidden Markov model (HMM). Experiments on twenty years worth of New York Times articles demonstrate the usefulness and efficiency of our approach.

Categories and Subject Descriptors
[H.3.3] **Information Search and Retrieval** Query formulation, Search process

General Terms
Algorithms, Experimentation

Keywords
Terminology evolution, web archives

1. INTRODUCTION

Web archives play a seminal role in preserving our cultural heritage for future generations. Among them, there are efforts such as the Internet Archive [1], which has been archiving the publicly-accessible Web for more than a decade, but also other long-term document archives such as those operated by newspaper companies. These archives constantly grow in size as the Web evolves and new content is created, but also thanks to improved digitization techniques, which make it possible to add content that was originally published a long time ago. As a consequence, documents archived in these vast collections now cover at least decades, sometimes even centuries (e.g., the archives of the London Times range back until the late 18th century).

When searching these long-term archives, one difficult problem arises from the fact that terminology and general language use evolve constantly – as first identified in [22]. To illustrate this problem, consider the following two use cases.

- Carl Curious, a student of arts, is writing a thesis about museums in Europe and searches a web archive for background information by issuing the keyword query saint petersburg museums. Not knowing that the city of Saint Petersburg is formerly known as Leningrad, Carl does not see those old but highly-relevant documents published in the 1970s with details on the Hermitage in Leningrad, since these would not be retrieved by state-of-the-art retrieval methods that rely on keyword matching.

- Nelly Noise, a physician, is researching on hearing damage that can be caused by portable music players and issues the keyword query ipod hearing damage. Documents published in the 1980s that describe sudden deafness observed with heavy users of the Sony Walkman –the dominant portable music player at that time– would not be found.

As the two examples demonstrate, terminology evolution negatively affects retrieval effectiveness and thus user satisfaction, when searching web archives. This is because users typically employ current terminology when formulating queries – there is thus a widening gap between the terminology used in the queries and the terminology that was utilized in the past to write the now archived documents. Tackling this problem is essential in order to keep archived contents accessible and interpretable.

At a first glance, query expansion and refinement techniques [14], as often employed in Information Retrieval to deal with the word-mismatch problem between queries and documents, may seem like an adequate solution. These techniques modify the user's query, for instance, by adding highly correlated terms. This is insufficient in our case for three reasons. First, there is not necessarily a high correlation between terms that were actively used in the past (e.g., walkman) and their current counterparts (e.g., ipod). Second, good rewritings should consider entire phrases even if the user gives only keywords (without phrase delimiters); for example middleware project costs could be mapped to TP monitor man months for the 1980s. Third, independently

*Partially supported by the EU within the 7th Framework Programme under contract 216267 "Living Web Archives (LiWA)"

Twelfth International Workshop on the Web and Databases (WebDB 2009), June 28, 2009, Providence, Rhode Island, USA.

substituting individual words by correlated words or phrases may dilute the query's coherence and create an undesired topic drift; for example, rewriting afro american president into african US chairman loses the user intention.

For these reasons, we take a different approach in this work. Given the user's query, formulated using today's terminology, our aim is to identify query reformulations that aptly paraphrase the user's information need employing terminology prevalent in the past. For the two use cases above, we would thus present Carl and Nelly with queries such as leningrad hermitage and walkman deafness, respectively. Such query reformulations are insightful by themselves and can then be issued to retrieve old documents that are highly relevant to the user's information need.

Finding adequate query reformulations also poses efficiency challenges for two reasons. First, the large scale of the web archives that we operate on, which comprise at least millions but often billions of documents. Second, users are impatient, so that achieving interactive response times at the order of at most few seconds is mission critical.

Contributions made in this work include (i) a measure of across-time semantic similarity that assesses the degree of relatedness between two terms when used at different times, (ii) a query reformulation method that determines good query reformulations for a given user query, (iii) a detailed description of how this method can be implemented efficiently to achieve interactive response times, and (iv) an experimental study conducted on twenty years worth of New York Times articles demonstrating the usefulness and efficiency of our approach.

Organization. The rest of this paper is organized as follows. Our formal model and notation are introduced in Section 2. Section 3 delineates our measure of across-time semantic similarity. In Section 4 we describe our novel query reformulation method that avoids drifting towards incoherent queries. Section 5 details on how the method can be implemented efficiently. Our experiments described in Section 6 demonstrate the usefulness and efficiency of our approach. Finally, we discuss related prior research in Section 7, before concluding this work in Section 8.

2. MODEL

In this section we lay out our formal model and the notation employed in the remainder.

Collection Model. We operate on a *timestamped document collection* $\mathcal{D}$. Each document $d^t \in \mathcal{D}$ bears a timestamp t that conveys its publication time. Timestamps are drawn from a *time domain* $\mathcal{T}$. We employ a discrete definition of time – timestamps can thus be thought of as positive integers (i.e., $\mathcal{T} = \mathbb{N}^+$). The special value *now* always points to the current time and we assume $\forall t \in \mathcal{T} : t < now$. The techniques presented in the remainder are based on temporal partitions of the document collection. Given a time interval T, we consider all documents d^t that were published during T, i.e., we demand $t \in T$.

Collection Statistics. We let $\mathcal{V}$ denote the vocabulary of terms occurring in documents. Here, our notion of term includes keywords, but may also include multi-word expressions such as entity names. For a term $u \in V$ we let $u@T$ denote the term when used during the time interval T. When writing $u@T$ and $u@R$, for instance, we thus refer to the same term u used during different time intervals. The number of occurrences of u in documents published in the time interval T is denoted as $freq(u@T)$. Further, given two terms u and v and a time interval T, the number of co-occurrences of the two terms is denoted as $cooc(u@T, v@T)$. In practice, this last figure is subject to further constraints. Thus, we may count a co-occurrence only (i) if u and v are within the same sentence, (ii) within a window of size ω, or (iii) we may take their order into account.

3. ACROSS-TIME SEMANTIC SIMILARITY

Having laid out our model and notation, we next introduce a method to assess the semantic similarity between two terms when used at different times, which will be a crucial building block when reformulating queries.

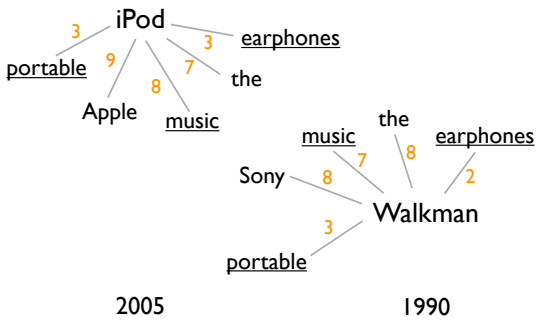

Figure 1: iPod@2005 and Walkman@1990 and their respective frequently co-occurring terms

How can we assess the semantic similarity between two terms when used at different times? As a running example, consider the two terms iPod@2005 and Walkman@1990, for which we would like to assess a high degree of semantic similarity, since both devices were the dominant portable music players at the respective time. Figure 1 shows the two terms with their respective frequently co-occurring terms. As apparent from the figure, simple co-occurrence between the two terms, as often used by query expansion techniques, is not helpful here – neither of the terms occurs frequently together with the respectively other term. Notice, though, the significant overlap between the terms that frequently co-occur with iPod@2005 and Walkman@1990 as, for instance, portable, music, and earphones. This significant overlap is a clear indication that the two terms are used in similar contexts at their respective time, which suggests the following:

Key Idea. The degree of across-time semantic similarity between two terms $u@R$ and $v@T$ can be assessed by comparing the contexts –captured by co-occurrence statistics– in which u and v appear at time R and T, respectively.

This key idea can be seen as an adaptation of the *contextual hypothesis* that has been around for decades [17, 20] and also served as a basis for recent work on using web search engines to assess semantic relatedness [6, 21].

In order to assess the degree of across-time semantic similarity, we propose a two-step generative model building on the above idea. In a first step, a term $w@T$ is randomly picked among terms that co-occur with the given $v@T$. Following that, in the second step, a term $u@R$ is picked among the terms that co-occur with the intermediate term w in documents published during R. In both steps, terms are chosen with probability proportional to the observed co-occurrence statistics for the respective time. As an intuition behind the model, consider a user trying to find terms that are used in similar contexts as $v@T$ – a natural way to do so is to

first identify terms that appear often together with $v@T$ (as portable, music, and earphones in our example), before examining their respective contexts to identify terms that appear often with all of them.

According to our generative model the probability of producing the term $u@R$ from the term $v@T$, which serves as our measure of across-time semantic similarity, is

$$P(\, u@R|v@T\,) = \sum_{w \in \mathcal{V}} P(\, u@R|w@R\,) \cdot P(\, w@T|\, v@T\,) \quad (1)$$

where $P(\, u@R \mid w@R\,)$ and $P(\, w@T \mid v@T\,)$ are estimated based on the available co-occurrence statistics as

$$P(\, u@R \mid w@R\,) = \frac{cooc(\, w@R,\, u@R\,)}{\sum_{z \in \mathcal{V}} cooc(\, w@R,\, z@R\,)}\ , \text{and} \quad (2)$$

$$P(\, w@T \mid v@T\,) = \frac{cooc(\, v@T,\, w@T\,)}{\sum_{z \in \mathcal{V}} cooc(\, v@T,\, z@T\,)}\ , \text{respectively.}$$

Note that terms that co-occur frequently with $v@T$ but are frequent in general, such as the in Figure 1, have little impact on the assessed degree of similarity. Since such generally frequent terms co-occur with many terms, for them the value $P(\, u@R \mid w@R\,)$ is low, so that their contribution to (1) is limited.

In practice, for a given $u@R$, one is often interested in efficiently determining the k terms $v@T$ having the highest degree of across-time semantic similarity. As we detail in Section 5 this can be accomplished efficiently using existing top-k query-processing techniques.

4. QUERY REFORMULATION

We now proceed to the core of this work and describe how queries can be reformulated to counter the negative effects induced by terminology evolution.

The problem addressed in this work can be formally stated as follows. Given a user query $q = \langle\, q_1, \ldots, q_m\,\rangle$ consisting of m query terms q_i, a reference time R, and a target time T, our aim is to identify a query reformulation $q' = \langle\, q'_1, \ldots, q'_m\rangle$ that aptly paraphrases the user's information need using the terminology valid at time T.

Notice that in the remainder of this section, for notational convenience, times R and T are omitted for terms in the original query and query reformulations. Whenever we write q_i to refer to a term in the original query, the corresponding time is implicitly assumed to be the reference time R. Analogously, when referring to a term q'_i in a query reformulation, the corresponding time is assumed to be the target time T.

What makes a query reformulation q' one that aptly translates the user's information need? Consider again our example query saint_petersburg museum@2005, for which we would like to determine good query reformulations for the target time $T = 1990$. By means of this example, we next identify three desiderata for query reformulations.

Similarity. As mentioned earlier, a good query reformulations, for our example query would be leningrad museum@1990. Notice that individual query terms in this reformulated query have a high degree of across-time semantic similarity with their counterparts in the original query. This would not be true for most other potential query reformulations such as economy europe@1990, for which the across-time semantic similarity between terms and their counterparts in the original query will be much lower. Thus, a first desideratum for a reformulated query q' is that its query terms are across-time semantically similar to their counterparts in the original query q', i.e., we aim for high values of $P(\, q_i \mid q'_i\,)$.

Coherence. When taking into account only this first desideratum, though, we may end up with a nonsensical query reformulation. Consider leningrad smithsonian@1990 as such a nonsensical reformulated query. This query meets our first desideratum, since its query terms leningrad@1990 and smithsonian@1990 are semantically similar to their counterparts saint_petersburg@2005 and museum@2005, respectively. However, putting together the two terms leningrad and smithsonian makes little sense, given that the Smithsonian Institution, which comprises different museums, is located in Washington D.C. As this example shows, it is important to assert that putting the query terms q'_i next to each other makes sense, or to put it differently, whether the terms contained in the reformulated query are *coherent*. One way to do so is to examine whether the query terms co-occur frequently at the target time T, which can be done using the co-occurrence statistics that are at our disposal. Thus, as a second desideratum, we aim for high values of $P(\, q'_i \mid q'_{i-1}\,)$ to assure a high level of coherence.

Popularity. Although, similarity and coherence as argued above are crucial desiderata, when determining good query reformulations, they still do not suffice. As an illustrating example, consider the reformulated query saarbruecken saarland_museum@1990. This query reformulation is reasonable with regard to similarity, since both Saarbruecken and Leningrad are cities and the Saarland Museum is a local museum. Also, with regard to coherence, the reformulated query is fine, given that the two terms saarbruecken and saarland_museum appear frequently together. However, it is unlikely, that this query reformulation is a satisfying reformulation that captures the user's information need, which could be to find about museums in large European cities. Therefore, we should take into account how often query terms in the reformulated query occur at the target time, to avoid constructing whimsical query reformulations as the one above. To this end, we aim for terms q'_i in the reformulated query that occur frequently, thus having a high value $P(q'_i@T)$, which is defined as

$$P(u@T) = \frac{freq(\, u@T\,)}{\sum_{z \in \mathcal{V}} freq(\, z@T\,)}$$

for a term u and time T.

Now that we have identified the three desiderata *similarity*, *coherence*, and *popularity*, we next describe our approach to across-time query reformulation, which is based on a hidden Markov model (HMM). For a general introduction to HMMs and their uses in natural language processing the reader is referred to Manning and Schütze [15].

Our HMM can be seen as a random process that generates queries using terminology prevalent at the reference time R. The alphabet of emittable symbols thus consists of all terms $v@R$. The state space of our HMM comprises all terms $v@T$. The initial state probability for the state $v@T$ (i.e., the probability to start in that state) is $P(\, v@T\,)$ as described above; it depends on the term's frequency of occurrence in documents published during T, and factors in the desideratum of popularity. When entering a state, a symbol is emitted – for the state $v@T$ the probability of emitting the symbol $u@R$ is $P(\, u@R \mid v@T\,)$, which is the across-time semantic similarity defined earlier. Having emitted a symbol, a transition is made to the next state. The probability of making

a transition to $w@T$ when being in state $v@T$ is defined as $P(\, w@T \mid v@T\,)$, which depends on how often the terms w and v appear together in documents published during T, and factors in our requirement of coherence.

Notice that query reformulations $q' = \langle\, q'_1, \ldots, q'_m \rangle$ correspond to state sequences in the above HMM. Good query reformulations according to our model can now be determined as those whose corresponding state sequence has a high probability of being traversed while emitting the original query $q = \langle\, q_1, \ldots, q_m \,\rangle$. Formally this probability is given as

$$P(\, q \mid q'\,) = P(q'_1) \cdot P(q_1 \mid q'_1) \cdot \prod_{i=2}^{m} P(q'_i \mid q'_{i-1}) \cdot P(q_i \mid q'_i)\ . \quad (3)$$

The best-k query reformulations can be determined in the following manner. In a first phase the well-known Viterbi algorithm [15] is run. Using dynamic programming, the Viterbi algorithm determines for each state the maximal probability of being in the state after i steps have been performed. Leveraging the information memoized by the Viterbi algorithm, in a second phase, an A* search is performed to determine the k state sequences having the highest probability of being traversed while emitting the original query. For a detailed description of this computation, we refer to Federico and Bertoldi [10]. The time complexity of the algorithm is in $O(\, m \cdot |V|^2\,)$ its space complexity is in $O(\, m \cdot |V|\,)$. This may seem prohibitive first, given that $|V|$ is typically at the order of 10^7 and that we aim at interactive response times. Fortunately, though, for a given query q, large portions of the HMM can be disregarded during the computation, as we describe in more detail in the Section 5.

5. IMPLEMENTATION

So far, we have paid only little attention to how our methods can be implemented efficiently as to achieve our objective of interactive response times.

Pruning the State Space. As we explained in the previous section, the time and space complexity of the algorithm that we use to determine good query reformulations crucially depend on the number of states in our HMM. Fortunately, many of the states do not influence the result and can therefore be ignored during the computation. Thus, at query-processing time, we only have to consider a small part of the HMM that is sufficient to compute the accurate result. In detail, we can ignore all states corresponding to terms $v@T$ fulfilling

$$\forall q_i \in q\ :\ P(\, q_i@R \mid v@T\,) = 0\ . \quad (4)$$

These states correspond to terms that do not emit any of the original query terms and can be safely ignored. This is because, by (3), state sequences that include such a state have zero probability of generating our original query.

Our implementation allows pruning the state space even further. In detail, for each of the original query terms $q_i@R$ we only consider the κ terms $v@T$ having highest probability of emitting the original query term. By definition of our across-time semantic similarity measure, these κ terms can identified efficiently using using top-k query processing techniques such as the family of TA algorithms proposed by Fagin et al. [9] – details are omitted due to lack of space. However, in contrast to the pruning condition given above, this additional pruning is not safe and thus produces only an approximate solution.

Precomputations. In order to speed up the computation of good query reformulations, we precompute values $P(u@T)$ and $P(\, u@T \mid v@T\,)$ for a fixed set of times T, corresponding to calendar years, and keep them in main memory.

Computing Query Reformulations. At query-processing time, good reformulations for a given query are then efficiently determined as follows. First, for each of the original query terms, we identify the κ (typically $1,000$) terms $v@T$ to be included in the state space, as described above. Having built up the relevant portion of the state space, the Viterbi algorithm is run. Our implementation looks up values $P(\, u@T \mid v@T\,)$ economically based on the rationale that we can spare looking up $P(\, u@T \mid v@T\,)$, if the state $v@T$ has zero probability of being visited. This is opposed to eagerly looking up values $P(\, u@T \mid v@T\,)$ for all term combinations. Finally, the best-k query reformulations are determined using A* search as described.

6. PRELIMINARY EXPERIMENTS

To evaluate the usefulness and efficiency of our approach, we conducted a preliminary series of experiments that is discussed in this section.

6.1 Dataset & Setup

Dataset. As a dataset we use the recently released New York Times annotated corpus [2]. This dataset contains more than 1.8M newspaper articles published between 1987 and 2007.

We further enriched the dataset, by annotating common phrases using the following "poor man's" phrase extraction technique. For all term sequences consisting of up to eight terms and matching the title of an article in the English Wikipedia, we add a special term to the document that represents the phrase. The rationale here is that by annotating common phrases, we get a hold on entity names, slogans, and other multi-word expressions. As an example, if a document contains the phrase "john lennon", we add the special term john_lennon to the document, since there is a corresponding Wikipedia article.

Collection statistics were precomputed for temporal partitions corresponding to calendar years. For the co-occurrence statistics we employ a value $\omega = 10$, take into account sentence boundaries, and disregard term order, i.e., whenever two terms u and v appear in the same sentence less than 10 terms apart, we count it as one co-occurrence. To remove noise and reduce the size of the data, for each temporal partition, values $cooc(\, u@T,\, v@T\,)$ smaller than 5 are removed, i.e., we consider only pairs of terms u and v that occur at least five times together in documents published during T.

Setup. We implemented all methods in a small prototype system using Java 1.6 as a programming language. Data (including co-occurrence statistics, term frequencies etc.) was kept in an Oracle 10g relational database. The experiments described below were run on a single SUN server-class machine having four AMD Opteron single-core CPUs, 16GB RAM, a large network-attached RAID-5 disk array, and running Microsoft Windows Server 2003.

6.2 Across-Time Semantically Similar Terms

Across-time semantic similarity, as introduced in Section 3, plays a central role in our approach. Therefore, in this first part of our experimental evaluation, we examine how much sense the terms considered to have high across-time semantic similarity make. For each of the five terms pope_benedict,

u / R/T	**pope_benedict** 2005 / 1990	**starbucks** 2005 / 1990	**linux** 2005 / 1990	**mp3** 2005 /1990	**joschka_fischer** 2005 / 1995
1.	alexander_pope	dunkin_donuts	unix_operating_system	audio_cd	klaus_kinkel
2.	the_pope	dunkin	unix_systems	digital_audio	klaus
3.	cardinal_ratzinger	donuts	unix_international	computer_files	bobby_fischer
4.	joseph_cardinal_ratzinger	coffee_shops	the_operating_system	s_files	stanley_fischer
5.	pope_john_paul	cup_of_coffee	disk_operating_system	the_rockford_files	searching_for_bobby_fischer
6.	pope_john_paul_ii	a_cup_of_coffee	dos_operating_system	rockford_files	boris_spassky
7.	conservative_catholics	coffee_cup	operating_system	audio_system	german_foreign_minister
8.	polish-born	coffe_shop	operating_systems	audio_tapes	kinkel
9.	irish_catholics	morning_coffee	os	audio_equipment	chinese_foreign_minister
10.	frantisek_cardinal_tomasek	coffee_filter	os_2	audio_clips	foreign_affairs_minister_of_israel

Figure 2: Terms reported as most across-time semantically similar

starbucks, linux, mp3, joschka_fischer, Figure 2 shows the ten terms considered most across-time semantically similar for the respectively specified reference and target time.

Consider the term pope_benedict in the second column with a reference time $R = 2005$ and target time $T = 1990$. It is noteworthy that our method both identifies terms as similar that (i) relate to Pope Benedict's former name Joseph Ratzinger, but also (ii) to the Pope John Paul II who was pope in 1990.

Also, for the term mp3 with analogous reference and target time, terms relating to other music media such as audio_cd and audio_tapes are among the identified terms. However, also misleading terms such as rockford_files, which refers to a TV drama, are reported – because these terms are also often used in context with terms such as files.

Finally, for the term joschka_fischer with a reference time $R = 2005$ and target time $T = 1995$ are shown. Our method brings up terms related to Klaus Kinkel, the German foreign minister in 1995, and other foreign ministers, which makes sense given that Joschka Fischer was foreign minister in 2005. Again, some of the terms are misleading and relate to chess player Bobby Fischer – because of a strong connection through the common last name and thus frequent co-occurrence with fischer.

6.3 Query Reformulation Results

In this second part of our experimental evaluation we examine the quality of query reformulations produced by our method. For each term q_i in the original query we consider the up to $\kappa = 1,000$ terms having the highest probability of emitting q_i. Figure 3 shows the query reformulations produced by our method for nine different queries using the reference time $R = 2005$ for all of them. When determining the best query reformulations using our method, we filter out query reformulations that are redundant in the sense that one of the query terms is a substring of another query term. These are rare but occur occasionally as an artifact of our corpus enrichment by phrase extraction. Notice that this filtering does not affect the algorithm, but can be done efficiently during the A* search phase. Response times when computing the query reformulations presented were in the order of 3–7 seconds, when using the system setup described above – a more systematic evaluation of this aspect and further optimizations of the implementation are part of our ongoing work.

Consider the first query george_bush speech, for which a target time $T = 1990$ was employed. In this case, the first query reformulation, which actually does not change the query at all, makes sense, considering that at that time George H. W. Bush was in office. Given the fact that Barbara Bush –the first lady at the time– gave the comencement speech at Wellesley College in 1990, also the query reformulations relating to her are sensible.

For the query yahoo acquisition, as the second query discussed here, and a target time $T = 2000$ our method produces query reformulations most of which pertain to mergers and acquisitions among technology companies that happened during this period.

Finally, as the last of our queries discussed in detail, consider the query tony_blair prime minister for which we used a target time $T = 1990$. The query reformulation margaret_thatcher prime minister, considered best by our method, relates to Margaret Thatcher, Great Britain's prime minister at that time. Apart from that, three of the remaining query reformulations relate to the prime ministers of Israel and Japan at that time, respectively.

Summary. The anecdotal results presented clearly show that our methods make an important step in the right direction toward countering negative effects induced by terminology evolution. Admittedly, as can also be seen from the results presented, there is room for future refinement. For instance, for the query reformulations produced for the query colin_powell iraq, a more *diverse* set of query reformulations may have been preferable.

7. RELATED WORK

Before concluding this work, let us put it in context with existing prior research, which can broadly be categorized as follows

Query Expansion and Query Refinement. Query expansion techniques [7, 23, 25, 24, 5, 8, 19] address the so-called *word-mismatch* problem in Information Retrieval. To this end, the initial query is extended with terms that have been observed to co-occur often with the query terms (i) in the corpus as a whole (global techniques) or (ii) in a set of documents relevant to the initial query (local techniques). However, there are two key difference that distinguish existing query expansion and refinement techniques from our work. First, in their setting times is not explicitly taken into account, so that negative effects as the ones mentioned in the introduction can not be alleviated. Second, these techniques are typically implemented to be transparent to the user. Therefore, since the expanded query is not presented to the user, it is less critical if the query becomes unintuitive to the user or incoherent (e.g., by generating query expansions with several tens of keywords, but using a non-conjunctive relevance scoring model).

Cross-Language Information Retrieval. Prior work in cross-language information retrieval has addressed the question of how a user query can be translated from one language to another [4, 10, 11, 13, 18]. Closest to our work, Federico and Bertoldi [10] employ a HMM for query translation. Although technically similar, there are two important differences that distinguish this line of research from

q / R/T	**george_bush speech** 2005 / 1990	**colin_powell iraq** 2005 / 1990	**kyoto protocol** 2005 / 1990
1.	george_bush speech	james_baker saddam_hussein	berenter greenhouse
2.	president_ronald_reagan excerpts	james_baker hussein	greenhouse_effect warming
3.	barbara_bush comencement	james_baker iraq	greenhouse_effect gases

q / R/T	**yahoo acquisition** 2005 / 1995	**christo gates** 2005 / 1995	**nintendo ds** 2005 / 1990
1.	telesis sbc	jeanne-claude christo	game_boy nintendo
2.	time_warner merger	christo reichstag	video-game nintendo
3.	america_online merger	christo the_reichstag	galoob nintendo

q / R/T	**tony_blair prime minister** 2005 / 1990	**angela_merkel berlin** 2005 / 1995	**airbus a380** 2005 / 2000
1.	margaret_thatcher prime minister	kohl helmut	airbus industries
2.	yitzshak_shamir prime minister	christian_democratic_union kohl	a3xx superjumbo
3.	vacek minister prime	helmut kohl	airbus superjumbo

Figure 3: Query reformulations

our work. First, in CLIR the existence of a dictionary is assumed that provides accurate translation for each of the original query terms – a task for which we have to resort to our measure of across-time semantic similarity. Second, as discussed above for query-expansion techniques, their focus is not on producing queries that make sense to the user.

Suggesting Alternate Queries. Query suggestion is commonly used by current web search-engines to help users to formulate their queries with less effort. In addition, these alternate query formulations are also very useful in the sponsored search domain - as a way to identify paid ad results that could placed in the result rankings. Most of the state-of-the-art methods in this setting utilize the large scale query log and/or click-through data to derive co-occurrence statistics and query reformulation behavior of users within a session [12, 16, 3]. Unfortunately, for the problem addressed in this paper, we not only suffer from the paucity of query logs but also from the potential terminological variations within query logs.

8. CONCLUSION

In this work we have made a first step toward countering the negative effects that terminology evolution induces in web archive search. We have proposed a novel measure of across-time semantic similarity, which is useful beyond the application considered in this work. Apart from that, we have developed an efficient technique to reformulate user queries. Our experimental evaluation on a large-scale real-world dataset demonstrates the usefulness of the proposed techniques and their efficiency.

Future Research. There is ample room for future research. Our method always produces query reformulations having the same number of terms as the original query. Replacing query terms one by one may not always be appropriate – consider the query compaq company history@1990 as an example, which would ideally be reformulated into hewlett packard company history@2009. In our concrete implementation, we employed phrase extraction techniques as a workaround to this problem. Further, we assumed a fixed set of temporal partitions for which co-occurrence statics have been precomputed. Relaxing this assumption as to allow for temporal partitions specified in an ad-hoc manner poses significant efficiency challenges, which we plan to address in the future.

9. REFERENCES

[1] Internet Archive. http://www.archive.org.

[2] New York Times Annotated Corpus `http://corpus.nytimes.com`.

[3] I. Antonellis et al. Simrank++: Query Rewriting through Link Analysis of the Click Graph. In *WWW*, 2008.

[4] P.-Y. Berger and J. Savoy. Selecting automatically the best query translations. In *RIAO*, 2007.

[5] B. Billerbeck et al Query expansion using associated queries. In *CIKM*, 2003.

[6] R. L. Cilibrasi and P. M. B. Vitanyi. The Google Similarity Distance. In *IEEE TKDE*, 19(3), 2007.

[7] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *CIKM*, 2005.

[8] H. Cui et al. Probabilistic query expansion using query logs. In *WWW*, 2002.

[9] R. Fagin et al. Optimal aggregation algorithms for middleware. In *JCSS*, 66(4), 2003.

[10] M. Federico and N. Bertoldi. Statistical Cross-Language Information Retrieval using N-Best Query Translations. In *SIGIR*, 2002.

[11] R. Hu et al. Web query translation via web log mining. In *SIGIR*, 2008.

[12] R. Jones et al. Generating Query Substitutions. In *WWW*, 2006.

[13] Y. Liu et al. A maximum coherence model for dictionary-based cross-language information retrieval. In *SIGIR*, 2005.

[14] C. D. Manning et al. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[16] Q. Mei et al. Query Suggestion using Hitting Time. In *CIKM*, 2008.

[17] G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1), 1991.

[18] C. Monz and B. Dorr. Iterative translation disambiguation for cross-language information retrieval. In *SIGIR*, 2005.

[19] Y. Qiu and H.-P. Frei. Concept based query expansion. In *SIGIR*, 1993.

[20] H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *CACM*, 8(10), 1965.

[21] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *WWW*, 2006.

[22] N. Tahmasebi et al. Terminology evolution in web archiving: Open issues. In *IWAW*, 2008.

[23] M. Theobald et al. Efficient and self-tuning incremental query expansion for top-k query processing. In *SIGIR*, 2005.

[24] O. Vechtomova et al. Query expansion with long-span collocates. In *Inf. Retr.*, 6(2), 2003.

[25] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *SIGIR*, 1996.