Ad Hoc Monitoring of Vocabulary Shifts over Time

Tom Kenter[†] tom.kenter@uva.nl
Melvin Wevers[‡] m.j.h.f.wevers@uu.nl
Pim Huijnen[‡] p.huijnen@uu.nl
Maarten de Rijke[†] derijke@uva.nl

[†]University of Amsterdam, Amsterdam, The Netherlands
[‡]University of Utrecht, Utrecht, The Netherlands

ABSTRACT

Word meanings change over time. Detecting shifts in meaning for particular words has been the focus of much research recently. We address the complementary problem of monitoring shifts in vocabulary over time. That is, given a small seed set of words, we are interested in monitoring which terms are used over time to refer to the underlying concept denoted by the seed words.

In this paper, we propose an algorithm for monitoring shifts in vocabulary over time, given a small set of seed terms. We use distributional semantic methods to infer a series of semantic spaces over time from a large body of time-stamped unstructured textual documents. We construct semantic networks of terms based on their representation in the semantic spaces and use graph-based measures to calculate saliency of terms. Based on the graph-based measures we produce ranked lists of terms that represent the concept underlying the initial seed terms over time as final output.

As the task of monitoring shifting vocabularies over time for an ad hoc set of seed words is, to the best of our knowledge, a new one, we construct our own evaluation set. Our main contributions are the introduction of the task of ad hoc monitoring of vocabulary shifts over time, the description of an algorithm for tracking shifting vocabularies over time given a small set of seed words, and a systematic evaluation of results over a substantial period of time (over four decades). Additionally, we make our newly constructed evaluation set publicly available.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing; H.3.3 [**Information Search and Retrieval**]: Retrieval models

Keywords

Vocabulary shift; distributional semantics

1. INTRODUCTION

Word meanings change over time [11, 27]. Detecting shifts in meaning for particular words has been the focus of much research recently [11, 13, 14, 18, 19, 29]. In this paper we address the complementary problem of monitoring shifts in vocabulary over time. Rather than taking a word as an anchor to monitor its (shifts in) meaning over time, we take the meaning as an anchor, and monitor the evolving set of words that are used to denote it. As an example, consider music storage media. Nowadays, we carry music with us on iPods and mp3 players. Before that there were compact discs. Prior to cds there were records, and music cassettes. Few of the words that were used in, say, the 1950s to describe the media used for storing music are still in use today. Following this example, the question we set ourselves to answer is "what words were used previously, where nowadays the words 'mp3 player' and 'iPod' are used?" An algorithm for monitoring vocabulary shifts over time has the words "iPod" and "mp3 player" as its input; as output it produces ranked lists of words per time period, e.g., every 5 years, of the words in that period that represent the concept underlying the initial input words. In what follows, we refer to such ranked lists of words per time period as *vocabularies*. The initial set of input terms we call *seed terms*.

Not all concepts evolve as dramatically as the sound carrier example above does, where the entire vocabulary changes in the course of a few decades. Often, many terms in the vocabulary remain relevant over time. A successful system for monitoring vocabulary shifts over time should strike a balance between an adaptive strategy that responds to changes in vocabulary, and a more conservative approach that keeps the vocabulary stable.

The problem setting we address is inspired by collaborations with digital humanities scholars in the field of history. Changes in discourse over time are a popular topic of studies in the humanities [10, 12, 15, 23]. Lists of keywords are usually maintained manually. However, "[f]inding the right keywords demands expert knowledge of the field of study and a great deal of perseverance and creativity" [15]. The methods for finding shifts in vocabulary over time that we propose in this paper are aimed at automating this task in a time-aware fashion. The resulting vocabularies are returned to the humanities scholars, as an indication of changes in discourse in the underlying corpus. They may be used for exploratory ends, to discover unfamiliar relevant historical terms. Additionally, as discussed in our future work section §6, if the vocabularies are of sufficient quality, they can be used for time-aware query expansion in an document retrieval setting for an historical corpus.

There has been extensive work on the related but different problem of concept drift in the context of ontologies and taxonomies; see, e.g., [27]. Any semantic ontology of terms should adapt over time in order to keep up with changes in meaning of the terms it contains. In this paper, however, we approach concept drift from a user perspective and not from an ontology perspective. This means that we do not assume pre-defined ontologies to be available and we do not aim to infer ontologies. Our primary motivation is to

CIKM'15, October 19–23, 2015, Melbourne, Australia.

ACM 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806474.

track evolving vocabularies over time for a user-provided topic of interest. This motivation leads to the following set of requirements:

1. **Words as retrieval unit** – Rather than outputting documents, as in a classic information retrieval scenario, an algorithm for monitoring shifts in vocabulary over time should, given a set of seed terms and a corpus, outputs words for a sequence of periods in time.
2. **Ad hoc** – An algorithm for monitoring shifts in vocabulary over time should work ad hoc. I.e., it should not be dependent on a predefined ontology or a fixed set of topics. The user should be able to provide ad hoc input at runtime. This requirement entails that very limited input of the user should be enough. Typically, one or two initial terms should suffice as an initial seed set.
3. **Broad time coverage** – An algorithm for monitoring shifts in vocabulary over time should be able to cover a substantial amount of time, at least multiple decades, long enough for interesting changes in discourse to occur.
4. **Comprehensible outcome** – The output produced by an algorithm for monitoring shifts in vocabulary over time should be easy to consume by humans. This means that the vocabularies that an algorithm yields should be limited in size, typically consisting of only a few words.

We note that an additional implication of the ad hoc requirement (requirement 2 in the list) is that no in-depth historical or domain knowledge of a user should be necessary. I.e., a user should not be required to have extensive knowledge of the concepts the input words are about nor of the underlying corpus. Rather, an effective method for monitoring shifts in vocabulary over time should provide new insights about the concepts and the corpus.

The comprehensible outcome requirement (requirement 4) entails that an optimal rate should be found for emitting vocabularies, regardless of a method's internals. If the rate is too low, too many vocabularies are produced, which leads to too much repetition. A rate that is too high would cause interesting shifts to go unnoticed. Precursory discussions with domain experts in the area of the history of ideas revealed that five years periods were deemed optimal.[1]

We propose an algorithm for monitoring shifts in vocabularies over time given a small user-provided set of seed terms and a period of reference. We note that this task is related to, but different from, tracking topics over time [9, 28], where topic models such as LDA and PLSA are used to monitor changes in a predefined number of topics. A crucial difference between topic modelling approaches and the method we propose is that, rather than relying on a predefined number of fixed topics, we allow for ad hoc queries.

Briefly, our proposed algorithm proceeds as follows. We first use distributional semantic models to infer a series of semantic spaces over time from a large body of time-stamped textual documents. We then construct semantic networks of terms based on their representation in the semantic spaces and use graph-based measures to calculate saliency of terms. Finally, we output shifting vocabularies over time—i.e., for a small set of seed words we output ranked lists of terms for a consecutive series of periods in time. The words in the vocabularies are meant to denote the same concept as the seed words do. As there is, to the best of our knowledge, no evaluation set available that allows for the intrinsic evaluation of monitoring shifts in vocabularies over time, we construct our own.

Our main contributions are:

- We introduce of the task of ad hoc monitoring of vocabulary shifts over time;
- We describe an algorithm for monitoring shifts in vocabularies over time, given a small, ad hoc set of seed words;
- We perform a systematic intrinsic evaluation of results of our proposed algorithm over a substantial time period (over four decades);
- We share our evaluation data, which can be downloaded from `http://ilps.science.uva.nl/resources/shifts`.

In the next section, §2, we discuss related work. In §3 we describe our method of tracking vocabularies over time. Our experimental setup is detailed in §4 while the results of the experiments are presented and analysed in §5. In §6 we conclude.

2. RELATED WORK

In this section we describe previous research related to the various aspects of our method of monitoring shifts in vocabularies over time.

2.1 Change in vocabulary over time with topic models

Topic models, like LDA and PLSA have been used extensively to monitor topics over time, starting with seminal work in [6, 28]. In [12] topic models are used to model the history of scientific ideas through time. The setting is similar, but different to the one addressed here, as word distributions of topics are inferred from the entire dataset and vocabulary shift is not modelled directly. Rather, the changes over time are modelled as shifts in the probability distribution of topics over the years. A related setting is addressed in [9] where topics and vocabulary are monitored over time.

The most important difference between topic model-based approaches, such as the ones discussed above, and the method we present in this paper is that our approach allows for an ad hoc setting. Topic models aim to infer a fixed set of latent topics from a corpus. This is the case even when non-parametric methods are employed [7], for which the number of topics is not fixed but inferred from the data. The non-parametric models allow for more flexibility, but once the algorithm has ran, there is a fixed set of topics it inferred. The inferred topics can be investigated to see interesting patterns over time, but if the topic of interest to the user is not in the inferred set of topics, there is no way around this.

Evaluation, from the perspective of the topics, is typically extrinsic, rather than intrinsic. The top-10 words for a selection of topics is shown in [12] but not evaluated. In [9] perplexity of the inferred topics is used as evaluation metric.

2.2 History of ideas

In the humanities, changing vocabularies are researched as well, in the field of intellectual history or the "history of ideas". In the context of the history of ideas, a distinction is made between the *intension* of an idea and its *extension* in [4]. The intension is the meaning of a concept, the extension comprises its mentions: "The extension of [a] concept differs through time. When confronted with certain changes in extension in the data, one likely conjecture is that the meaning of the concept [...] has changed" [4]. In this paper we regard the words used to denote this meaning as its extension, rather than sentences or entire articles as in, e.g., [27]. Although the intension of a concept changes as its extension changes,

[1] We note that alternatively, the optimal rate of emitting vocabularies could be determined programmatically. In theory, it could even differ between sets of seed words. The evaluation of such an approach would require extra, non-trivial annotator effort and we consider it outside the scope of the present paper.

we assume that the intension changes gradually over time (e.g., the intension of the concept of nuclear weapons is relatively stable over time, while the names of particular instances, and the words used to refer to nuclear weapons might change over time as the techniques involved evolve). By monitoring shifts in vocabularies over time, we aim to monitor shifts in the extension of a concept. We assume that the intension of a concept is continuous enough over time to allow for such monitoring. By adhering to this assumption we follow e.g. Kuukkanen who introduces a distinction between the core of a concept and its margins when discussing conceptual change: "the main idea is that conceptual continuity requires the stability of the core of the concept, but not necessarily that of the margin, which is something that enables a description of context-specific features" [20]. While we do not explicitly model the core or margin of concepts, we do assume conceptual continuity.

2.3 Topic detection and tracking

The goal of topic tracking systems, given a stream of documents, is to extract documents from the stream which are relevant to a set of topics of interest. Topics, in this setting, are typically events [1] or entities [8]. As events and entities may evolve over time, many adaptive document filtering algorithms have been proposed [3, 16, 25]. A sliding window approach is used on a stream of documents in [25], a component we also use in our method of monitoring shifting vocabularies over time in §3.

Document filtering algorithms typically contain a profile of the events or entities they monitor in the form of a (weighted) list of words which can adapt over time. Maintaining such a profile is clearly analogous to the task addressed in this paper, although we aim to track the words that are used to describe the meaning of a concept, rather than events or entities. Furthermore, it is important to note that in our present setting of vocabulary tracking the aim is to list terms that are semantically very similar to one another, while in the document filtering case it is beneficial for a filtering profile to cover a range of aspects as diverse as possible concerning the event or entity in question.

2.4 Change of word semantics over time

Research on detecting semantic shifts for words has seen a surge of interest recently. In [18] word vectors are trained on a corpus spanning over more than a century, with word2vec [21]. The vectors are trained on an incrementally growing time window, rather than a sliding window as we propose to do here. Several examples are shown to illustrate that dramatic semantic changes over time can be detected by monitoring the distance between the word vector of word in the initial model, that contains the least recent documents, and the vector from models trained on the windows including more recent material. Similarly, in [29] words are monitored over centuries. A number of examples is presented that show that changes in meaning as well as additional meanings of words can be detected. In [11] semantic change between words is measured with a distributional semantics method. The Google Books Ngram corpus is used to construct co-occurrence vectors of words in two decades (the 1960s and the 1990s, which is roughly the same time frame we use in our experiments). The task is to detect whether or not words have undergone a drastic semantic change, and human annotators were asked to annotate for a hundred words whether or not their meaning changed over the decades. In [13, 14] co-occurrence statistics are used to find related words to a specific term, which are monitored to find the sudden shifts in meaning.

We should note that, though monitoring the shifts in meanings of words over time is very related to the setting in these papers, there is a key difference in what we are trying to achieve. To illustrate, consider the main example used in [18]: the word "gay". The meaning of this word shifted considerably over the last century. Rather than focussing on the word "gay" itself to monitor its shift in meaning, the question we ask is: what words came in its place? Apparently, the meaning of the word "gay" evolved, and it now (largely) means something else from what it used to mean. So, which terms took its place? Which terms were used in a later time frame, to denote the meaning that was previously referred to by "gay?" Our aim is to track the *concept underlying a particular set of seed words* (of which there can be more than one). Crucially, in our adapative approach for monitoring vocabulary shifts over time, we allow the original seed words to disappear completely. However, as the task in this work is related to the one addressed in, e.g. [13, 14], we construct our baseline accordingly.

2.5 Distributional semantics

Distributional semantic approaches are based on the intuition that words appearing in similar contexts tend to have similar meanings. Words are typically represented as vectors where the vectors incorporate information about the context. Recent advances in neural network language models have led to new ways of computing word vectors, where more training material can be leveraged than was previously feasible. In our experiments in §4 we use word2vec [21] to infer word vectors. Word vectors, also referred to as word embeddings, embed words in a semantic space. This means that the word vectors for words with a similar meaning are close in the semantic space they are embedded in.

We note that using word2vec is not the only way to get distributional semantic word representations. Methods based on co-occurrence have been used for tasks similar to ours as described above. An alternative more similar to word2vec is the GloVe algorithm [24]. We use word2vec in our experiments as it has proven to yield high-quality word embeddings [2, 22]. The same goes for the GloVe algorithm but it needs considerably more resources in terms of training time and memory consumption, which is a drawback given the large size of our corpus. There is no theoretical restriction, however, on the choice of distributional semantic model in the algorithms we propose in §3.

2.6 Methods of evaluation

The evaluation used to assess the quality of the approaches discussed above is frequently based on small number of positive examples [12–14, 18, 19, 29]. Following [11] we perform explicit intrinsic evaluation, where we ask human annotators to judge the quality of the output of our algorithms directly.

The research presented in this paper extends previous work in a number of ways. Firstly, rather than focussing on monitoring the change in meaning of particular words over time, as in e.g. [11, 13, 14], we monitor the underlying concept, by monitoring the set of words that is used to denote it over time. We describe an algorithm for constructing a semantic network of related terms and for maintaining this network over time. Secondly, we do not rely on a fixed set of topics or pre-defined ontologies, but allow for ad hoc input: a small set of input words, specified by the user. Thirdly, we perform systematic intrinsic evaluation of our methods for generating shifting vocabularies over time.

3. MONITORING SHIFTING VOCABULARIES THROUGH TIME

In this section we describe our algorithm for monitoring shifts in vocabulary over time. By *vocabulary* we mean a ranked list of unique terms.

3.1 Overview

Our algorithms for monitoring shifts in vocabulary over time use three components: *sliding time windows*, *generation algorithm* and *aggregation algorithm*.

We use time windows of multiple years in length (we experiment with 5 and 10 year time windows in our experiments in §5) and extract documents out of our corpus that were published within the time window. The window length is in years and every next window starts one year later than the previous window. If we use, e.g., ten-year windows, and the overall time period starts in 1950, we have a 1950–1959 window, a 1951–1960 window, etc. From the documents published within a time window we compute a semantic model (see §2.5). So we have one semantic model for each sliding window in time. The computation of the semantic models is a pre-processing step. It is done only once for a given corpus.

As mentioned in §1 when discussing requirement 4, the optimal period for outputting vocabularies is five years. However, the sliding windows are one year apart. Because of this, our method for constructing vocabularies over time comprises two algorithms. The first algorithm, which we refer to as the *generation algorithm*, outputs a series of vocabularies, one for each sliding time window, using a semantic network from the semantic model constructed from the documents in the time window. The second algorithm, which we refer to as the *aggregation algorithm*, aggregates over the vocabularies generated by the generation algorithm to produce the final vocabularies for the desired time period.

The generation algorithm uses graph-based measures to extract the most salient words from a semantic network for a given time window. The salient words are used as input to the next iteration of the algorithm. In short, the generation algorithm takes the original user-provided words as its input and *adaptively* updates this seed set by iterating over the sliding time windows.

Our algorithms for generating shifting vocabularies over time are completely unsupervised. No labelled training data is needed. Furthermore, no pre-defined ontologies are necessary. Only a large amount of unlabelled data is needed to derive word vectors from.

In what follows we describe three methods of generating shifting vocabularies over time. The *adaptive* method uses both the generation algorithm and the aggregation algorithm. The *non-adaptive* only uses the aggregation algorithm to aggregate over vocabularies generated from the sliding time windows. The *hybrid* method combines the vocabularies produced by the adaptive and non-adaptive methods. As the sliding time windows are used by all three methods, we first turn to discussing these.

3.1.1 Sliding time windows

As detailed in §2.2 the intuition underlying our model for monitoring shifts in vocabulary over time is that word meanings, and the semantic relations between words, shift gradually and continuously over time [4, 20, 27]. To make use of this continuum when constructing semantic models, we split the time period we are monitoring in multiple time windows, and calculate a semantic model from each of these windows. I.e., we extract all documents from the corpus that were published in the desired time window and train a word2vec model on their text contents.

To be sensitive to rapid changes, it would be beneficial to have short time windows. However, previous research has proven that the quality of the semantic models inferred by word2vec is higher when more training data is used [21]. We solve this conflict in requirements on the size of the training data for the semantic models by using *overlapping* time windows. By taking an extended period of time, we obtain a sufficient amount of data for constructing high-quality semantic models. As the windows are only one year apart from each other, changes in the semantic relations between words can be detected between subsequent models, while the vast majority of relations will remain stable, due to the overlap.

3.2 Adaptive method for generating shifting vocabularies over time

In this section we describe the generation algorithm and the aggregation algorithm, for our adaptive method of monitoring shifting vocabularies over time.

3.2.1 Generation algorithm; generating shifting vocabularies over time

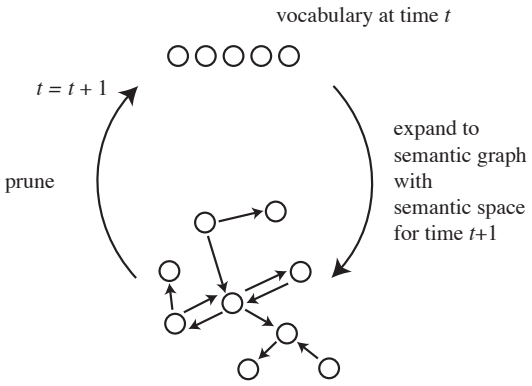

Figure 1: Schematic representation of the generation algorithm for generating vocabularies over time.

In Figure 1 a schematic overview is given of the generation algorithm for generating vocabularies over time from sliding windows. Every iteration consists of an expansion step and a pruning step. In the expansion step, a semantic graph is constructed from a list of seed terms and a semantic space. In the pruning step, the top terms, according to a graph-based measure, are extracted from the graph. This vocabulary is the input to the next iteration. As can be seen from this schematic overview, the original input words do not necessarily end up in the vocabulary one (or more) iterations later.

In Algorithm 1 the pseudocode for the generation algorithm is provided. At the very first iteration, the input consists of the seed set as provided by the user (Algorithm 1, line 1). As a key requirement of our method is limited effort by the user, we use only a few terms (typically one or two) as input. In the outer loop is carried out K times (line 2), once for every semantic model, derived from the K sliding time windows. In the expansion step (line 4–8), we construct a weighted, directed, partial semantic graph from the set of seed terms, given the semantic space from the next time window. To do this, we obtain the related terms for every word in the seed set, with a minimum similarity ς (line 5). Per seed set term we take at most n related terms. The terms obtained in this manner are the vertices of our graph. Form these vertices we construct a semantic graph (line 9). The edges in the graph are directed and weighted. We draw an edge from vertex w_i to w_j if w_j is in the list of related words of w_i. The weight on the edge is determined by the strength of the association between w_i and w_j in the semantic space. The network is partial as we do not construct an extensive network of all possible vertices (i.e., all word types in the corpus), but rather extract the part of the network in the vicinity of the seed terms. In the pruning step the top-n terms are selected relative to their degree centrality in the semantic network (line 10).

We use elementary variants of degree centrality: in-degree and out-degree. More involved measures like PageRank can be considered as well, especially when larger parts of the graph are extracted.

Required: $W = [w_1 \dots w_{|W|}]$: a set of seed terms
Required: Series of semantic spaces $S = [sem_1 ... sem_K]$, ordered by time
Required: ς: minimum similarity
Required: n: maximum number of terms to return
Result: List of vocabularies $v_1 \dots v_{|K|}$

1 $v_0 = W$;
2 **for** $k \leftarrow 1 \dots |K|$ **do**
3 vertices = [];
  /* expand */
4 **foreach** $w \in v_{k-1}$ **do**
5 **foreach** $w_{related} \in related\_words(w, sem_k, \varsigma, n)$ **do**
6 vertices = vertices $\cup$ $w_{related}$;
7 **end**
8 **end**
9 semanticNetwork = drawEdges(vertices);
  /* prune */
10 v_k = top-n nodes from semanticNetwork w.r.t. degree centrality
11 **end**

Algorithm 1: Generation algorithm: adaptively generating vocabularies from sliding time windows

e.g. by finding related words of related words, and so on. However, preliminary experiments showed that the relation between the original seed terms and related terms of related terms can quickly become arbitrary. A method relying on such terms would run a considerable risk of topic drift.

We compute four measures of degree centrality: number of inlinks, weighted sum of inlinks, number of outlinks and weighted sum of outlinks. The choice of degree centrality measure is a parameter of our model. We discuss the effect of this parameter on the results of our experiments in §5.2.

Direction in time. In this section we describe a forward pass, where we start with the oldest time window and progress towards the future. The same method can be applied the other way around, as would, e.g., be appropriate for the iPod example in §1. In Algorithm 1 this means that we start with $v_{|K|}$ in line 1, range over $k \leftarrow |K| \dots 1$ in line 2 and iterate over $w \in v_{k+1}$ in line 4.

3.2.2 *Aggregation algorithm: Producing the final output vocabularies*

For each semantic space, generated from documents in overlapping time windows one year apart, the generation algorithm generates a vocabulary. If we monitor, e.g., a period of four decades, 40 vocabularies are generated, one for every overlapping window. The final output presented to the user, however, should be one vocabulary for every 5 year period, so 8 vocabularies, in the example case. To generate the final output vocabularies, we aggregate over the vocabularies generated by the generation algorithm.

The aggregation step producing the final vocabularies is distinct from the principal underlying method of generating vocabularies for all overlapping time windows. If the final vocabularies should be generated for periods of 4 or 6 years, rather than 5, the output of the generation algorithm could be used unaltered, and only one parameter needs to be changed in the aggregation algorithm.

Algorithm 2 lists the pseudocode of our method for aggregating over the vocabularies output by Algorithm 1 to produce the final output vocabularies. The first step in each iteration (line 2) is to select a set of vocabularies relevant to the time period at hand τ_t. We select all vocabularies constructed from time windows that have an overlap with τ_t. This step is needed as the length of the time windows is a parameter of the model and might not be the same as the length of τ_i. We can, e.g., use 10-year windows in the generation algorithm, while we output vocabularies for 5-year periods in the final step (i.e. the length of every period τ_t is 5).

Required: List of vocabularies $V = v_1 \dots v_{|K|}$
Required: List of time frames $T = [\tau_1 \dots \tau_{|T|}]$ for which to output vocabularies
Required: n: maximum number of terms to return
Result: List of vocabularies $v_{\tau_1} \dots v_{|T|}$

1 **for** $t \leftarrow 1 \dots |T|$ **do**
2 $V' = [v \in V \mid v$ relevant to $\tau_t]$;
3 **foreach** $v \in V'$ **do**
4 **foreach** $w \in v$ **do**
5 $score_w += f_{weight}(v, \tau_t) * score_{w,v}$;
6 **end**
7 **end**
8 v_{τ_t} = top-n terms w sorted by $score_w$;
9 **end**

Algorithm 2: Aggregation algorithm: Aggregating vocabularies output by the generation algorithm to produce the final output vocabularies.

In the inner loops of Algorithm 2 we iterate over the words in the selected vocabularies (line 3–7). We compute a score for all words, which consists of their score in vocabulary v (their degree centrality, see previous section) weighted by a weight function $f_{weight}(v, \tau)$ that assigns a weight to a vocabulary v for a time frame τ.

Vocabulary weighting function. As described above, each vocabulary v_{τ_t} is constructed from a semantic space, derived from the texts of documents published in a time window, spanning a number of years. The time window has an overlap with time period τ_t that we want to output a vocabulary for. Therefore, a weighting is needed which expresses how much vocabulary v should contribute to v_{τ_t}, the final vocabulary we output for τ_t.

The most straightforward way of weighting is to weight all vocabularies equally (i.e., apply no weighting at all). However, the central years in the period the vocabulary is derived from are most likely to best capture its semantics (e.g., if we look at the decade 1970–1979, the documents in the early 1970s might still have echoes of the late sixties, while in the late 1970s, the 1980s might already become apparent; the middle years will define the vocabulary most clearly). We implement this intuition by assuming that the probability of the contribution of years to a vocabulary v is given by a Gaussian distribution, where the mean of the distribution is the centre of the period, and we assume a standard deviation of 1.0. We model the distribution of the years in τ in a similar fashion, where the mean is the central year of τ. Given these two distributions we can use the Jensen-Shannon divergence as a proxy for the weight of v with respect to τ:

$$f_{JSD}(v, \tau) = JSD(\mathcal{N}(\mu_v, \sigma_v^2) \parallel \mathcal{N}(\mu_\tau, \sigma_\tau^2)),$$

where we have $\sigma_v^2 = \sigma_\tau^2 = 1$.

We note that simple overlap metrics, like, e.g.. Jaccard distance, do not measure what we want, as a the Jaccard distance between two periods, where one period overlaps completely with the other, is always the same, regardless whether they overlap in the central region of the longer period or not.

3.3 Non-adaptive method for generating shifting vocabularies over time

Using the adaptive method for generating vocabularies, it is well possible that none of the words in the original seed set are present after a few iterations. This is a desired effect, but it also introduces the risk of topic drift. I.e., the adaptive algorithm might focus on an aspect of meaning that was not intended by the user, which can cause the vocabularies being generated to drift in the wrong direction. To counter this effect, we also include runs in our experiments where the initial seed set is kept static. That is, we omit Algorithm 1, and instead output the n words most related to the words in the original seed set for every sliding time window. To generate the final vocabularies we do employ Algorithm 2.

We refer to this method, that follows a static seed set for generating shifting vocabularies over time as *non-adaptive* method.

3.4 Hybrid runs

To combine the exploratory effect of the adaptive approach with the more conservative approach of the non-adaptive approach, we combine the runs of both methods of producing shifting vocabularies over time to produce *hybrid* runs. In particular, we replace the least central terms of the vocabularies produced by the non-adaptive method by the top i vocabulary terms produced by the adaptive method with respect to degree centrality. In §5 we report results for different values of i.

4. EXPERIMENTAL SETUP

To measure the quality of the different methods of generating vocabularies over time we perform a systematic, intrinsic evaluation. Our research questions are:

RQ1 Given that we have an exploratory, adaptive approach and a conservative, non-adaptive approach for generating shifting vocabularies over time, can we combine the two in such a way that performance is gained over the components?

RQ2 How do the parameters of the generation algorithm and the aggregation algorithm affect performance?

The first research question, RQ1, concerns the balance between an exploratory response to change in vocabularies, which introduces the risk of topic drift, and a static, conservative approach, which does not allow for substantial evolution of vocabularies. In §5.1 we report on the results for our experiments regarding this question.

The second research question, RQ2, concerns our algorithms for generating vocabularies over time more specifically. As detailed in §3 we construct semantic networks to find salient terms in specific time periods. We are interested in evaluating whether, e.g., the weighting of edges is beneficial or not, or whether selecting nodes based on in-degree yields better results than using out-degree.

We analyse the performance regarding all parameters of our algorithms of generating shifting vocabularies over time in §5.2. In the remainder of this section we detail the aspects of our experimental setup.

4.1 Ground truth data

The natural ground truth data for our task of monitoring shifting vocabularies over time are the shifting vocabularies themselves. We make use of human annotators to obtain this ground truth data. The annotators' task is, given all unique words occurring in a corpus of timestamped documents, to indicate which words are relevant to a particular topic of interest in a certain time period. As it is not feasible for annotators to judge all word types in a corpus, we employ a pooling approach, which we detail below. Below, we also provide the characteristics of the seed terms.

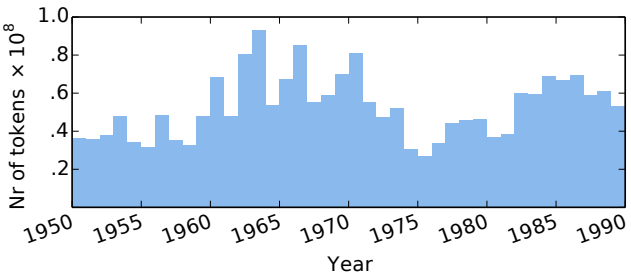

Figure 2: Number of tokens per year

Given a small number of seed terms, and a short text describing the underlying information need, the annotators were asked to judge terms per period on a 3 point scale: *irrelevant*, *related* and *perfect*. The *related* category is used for borderline cases in which a result is not completely off the mark, but is not exactly right either.

There were 6 annotators in total, all of whom are academic historians, well-acquainted with both the corpus and the evaluation time period. None of the authors took part in the annotation effort.

Following, e.g., [11] we use the pairwise Pearson correlation to determine inter-annotator agreement. The Pearson correlation coefficient is .555 with a p-value $< 10^{-5}$. It shows that the judgements are highly correlated between annotators and that the averaged judgements can reliably be used to evaluate our experiments.

The sets of seed terms and the ground truth annotations are publicly available. The material can be downloaded from `http://ilps.science.uva.nl/resources/shifts`.

Pooling. We produce output using each of the methods for generating vocabularies over time that we consider, and all combinations of parameters. We pool these results, similar to how the runs of IR systems are pooled in a classical TREC-style evaluation [26]. In our setting, however, the unit of retrieval is a word for a given time period, rather than a document. Annotators are presented with the aggregated results of all runs combined.

Corpus. Our corpus is a collection of Dutch newspapers, digitised by the Royal Library of the Netherlands.[2] We use four decades, 1950 up until 1990, as our evaluation period as this period is long enough for interesting changes to occur and modern enough for the OCR quality to be reasonable.[3]

The corpus contains 26 614 346 documents (newspaper articles) in the four decades we consider. Together, they comprise 1 940 841 unique words and 2 141 992 571 tokens. Figure 2 gives an overview of the numbers of tokens per year. As can be observed from this figure, the tokens are not evenly distributed across the years, but there is no bias towards either modern or historical documents. We used the Python NLTK Punkt Sentence Tokenizer [5] and remove additional unicode non-word characters.

Seed terms. There are 21 sets of seed terms in our experiments, which are provided by Dutch historians, who are familiar with the corpus and the time period selected. The terms are inspired by their own, real-life, research questions and by observations they

[2]The full newspaper corpus, and more, can be queried at `http://www.delpher.nl`.

[3]No official numbers concerning the OCR quality throughout this corpus are available. Anecdotal evidence suggests though that modern material is of higher quality.

Table 1: Overview of the Dutch seed set words

Seed words	English explanation	Direction
cd, compactdisc	cd, compact disc	backwards
computer	computer	backwards
doping	drugs (sports-related)	backwards
efficiency, efficiëntie	efficiency	backwards
gastarbeider, gastarbeiders, immigranten	immigrants	backwards
geboortebeperking, geboorteregeling	birth control	forwards
holocaust	holocaust	backwards
internet	internet	backwards
jodenvervolging, deportatie, deportaties	persecution of Jews (in WWII)	forwards
marxisme	marxism	forwards
multinational	multinational	backwards
neger, negers, negerin, kleurling	negro, coloured people	forwards
quiz	quiz	backwards
supermarkt	supermarket	backwards
waterstofbom, atoombommen waterstofbommen, atoombom	hydrogen bomb	forwards
zelfbedieningswinkel, zelfbedieningszaak, kruidenier	self-service shop	forwards
amsterdam, rotterdam, utrecht	large Dutch cities	forwards
boek, boeken, boekje	books	forwards
koe, koeien	cows	forwards
mozart, beethoven, brahms	classical composers	forwards
viool, violen	classical instruments	forwards

made from the corpus. As discussed in §3, an algorithm for generating vocabularies over time can run either forwards or backwards in time. It was left up to the historians to decide on the most natural direction in time, per set of seed terms. In Table 1 we present an overview of the seed sets, together with the direction in time. The bottom 5 rows in Table 1 list 5, so-called, a-historical seed sets. The concepts denoted by these seed sets are assumed, by the historians, to stay relatively stable over the entire evaluation period. We include the a-historical seed sets for two reasons. Firstly, we want to avoid a bias in the test set towards changing concepts, i.e., we do not want the test set to only consist of examples of which it is apparent that they evolve over time, as this would put the non-adaptive methods at an unfair disadvantage. Secondly, we want to check for over-generating, by which we mean, in this context, generating changing vocabularies while there is in fact no change. A method that is too exploratory might always find new terms and might show evolving list of words erroneously. To be able to measure such behaviour, we add the a-historical seed term sets.

On average the seed term lists are 2.1 words in length. The ground truth vocabularies (i.e. the list of relevant words per time period) are 9.32 words in length on average.

4.2 Evaluation

Our algorithms for generating shifting vocabularies over time produce ranked lists of words. The Cranfield-style evaluation setting allows us to use traditional IR evaluation metrics suitable for evaluating ranked lists, NDCG and MAP, in addition to the standard F_1 metric.

4.3 Parameters and settings

For the generation algorithm, we use 5-year and 10-year sliding time windows to compute semantical spaces from. Preliminary experiments showed that values between .6 and .7 are reasonable values for ς. Hence we experiment with $\varsigma \in [.6, .65, .7]$. For degree centrality we use 4 variants, as described in §3.2.1: sum of inlinks, weighted sum of inlinks, sum of outlinks, weighted sum of outlinks.

The aggregation algorithm has only one parameter: the vocabulary weighting function. We experiment with a uniform weighting function (i.e., no weighting), and the JSD-weighting function, described in §3.2.2.

As discussed in §2.5 we use word2vec [21] to generate word vectors for every time window. We employ default settings; Skip-gram architecture, with hierarchical softmax and no negative sampling, vector dimensionality of 300, window size of 5, and minimum word frequency of 5.

In all experiments, the vocabulary size n is set to 10.

4.4 Baseline

As noted in §2, the work described in [13, 14] is related to our present setting. Following this work, we construct our baseline by using a time slice approach. However, we use neural network language models to construct semantic models to derive semantic proximity from, rather than co-occurrence measures as in [13, 14], as the computation of a full co-occur-rence matrix on the corpus used in our experiments is intractable, due to its size. For every time window τ_t our baseline methods outputs the top-n most related words derived from a semantic model trained on the documents published in time window τ_t.

5. RESULTS AND ANALYSIS

We begin by answering our research questions and proceed by contrasting the adaptive approach and the non-adaptive approach, described in §3.2 and §3.3, respectively.

5.1 Hybrid vs. Non-hybrid approaches

To answer RQ1 we conduct experiments with all methods described above and all parameter settings. Table 2 contains an overview of the results yielded by the best parameter setting.[4]

Table 2: Results for JSD weighting with 10-year periods, $\varsigma = .65$, in-degree over weighted edges. Statistically significant differences from the baseline, measured with a two-tailed paired t-test, is marked for $p < .02^{\dagger}$ and $p < 10^{-6\ddagger}$

Method	F_1	p	r	NDCG	MAP
hybrid ($i = 1$)	$.384^{\ddagger}$	$.537^{\ddagger}$	$.406^{\ddagger}$	$.646^{\ddagger}$	$.343^{\ddagger}$
hybrid ($i = 2$)	$.391^{\ddagger}$	$.544^{\ddagger}$	$\mathbf{.414}^{\ddagger}$	$.650^{\ddagger}$	$\mathbf{.346}^{\ddagger}$
hybrid ($i = 3$)	$\mathbf{.392}^{\ddagger}$	$\mathbf{.548}^{\ddagger}$	$.411^{\ddagger}$	$\mathbf{.653}^{\ddagger}$	$.345^{\ddagger}$
hybrid ($i = 4$)	$.389^{\ddagger}$	$.545^{\ddagger}$	$.405^{\ddagger}$	$.651^{\ddagger}$	$.343^{\ddagger}$
hybrid ($i = 5$)	$.385^{\ddagger}$	$.541^{\ddagger}$	$.399^{\ddagger}$	$.649^{\ddagger}$	$.339^{\ddagger}$
adaptive	.344	$.551^{\ddagger}$	.298	$.514^{\dagger}$	$.237^{\dagger}$
non-adaptive	$.367^{\ddagger}$	$.521^{\ddagger}$	$.389^{\ddagger}$	$.630^{\ddagger}$	$.332^{\ddagger}$
baseline	.303	.450	.296	.554	.266

The key observation from Table 2 is that the hybrid approach outperforms both the baseline, and the adaptive method and non-adaptive method separately, on all metrics, regardless of the value of i. It is important to note that the parameter setting reported in Table 2 consistently yields the highest results on all metrics for the hybrid method, regardless of the value of i.

5.1.1 *Adaptive vs. non-adaptive*

The non-adaptive method outperforms the baseline by itself. The adaptive method only does so in terms of F_1. As is clear from Table 2, though, the adaptive method can add valuable information

[4]Note that due to macro-averaging, the macro-F_1 scores can and do end up lower than the individual macro-precision and macro-recall scores.

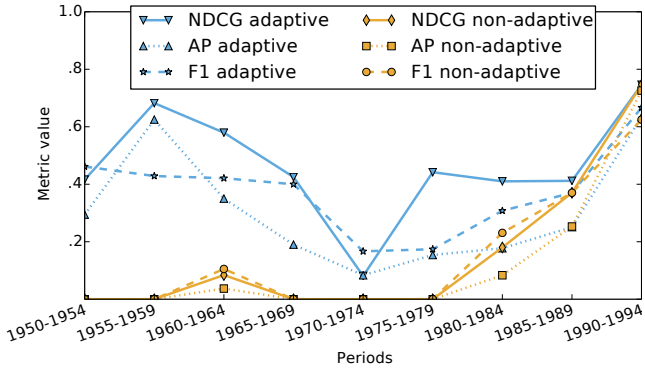

Figure 3: Comparison of results between the adaptive and non-adaptive run for the seed words "cd, compact disc." Direction is backwards in time.

to the non-adaptive method. In this section we present a number of examples to illustrate the difference between the two. To highlight the difference, we only show examples of the non-hybrid runs in this section. These runs contributed to the results in the rows labeled 'adaptive' and 'non-adaptive' in Table 2.

In Figure 3 the results are displayed for the non-adaptive run and the adaptive run for the seed words "cd, compactdisc." The direction for this example is backward, i.e., we start with the seed words in the 1990–1994 period and go backward in time.

As we can clearly see from the figure, the performance of the non-adaptive run quickly degrades over time (recall that we are going backward in time). Interestingly, the adaptive run, after a glitch in the 1970–1974 period, manages to pick up to get decent performance again for the time periods in the 1950s and 1960s. This indicates that the network approach, in which a network of related terms is promoted, can be beneficial.

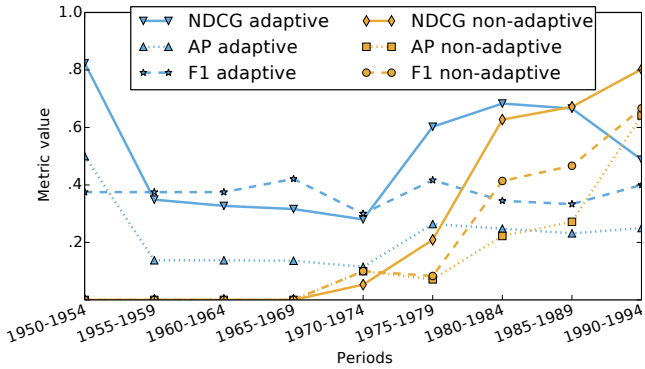

Figure 4: Comparison of results between the adaptive and non-adaptive run for the seed word "holocaust." Direction is backward in time.

We see a similar pattern in the results for the seed word "holocaust" in Figure 4. Again, we are going backward in time for this example. The performance of the non-adaptive run steadily degrades as we go back in time. This can be explained by the fact that the word "holocaust" barely occurs in the corpus prior to 1978.[5] The term was introduced in Dutch discourse by an American television series by that name. Initially, the term was used primarily to refer to the series, but gradually it became a more general term that now means the same as it does in English.

In Figure 5 the results are displayed for the seed word "multinational." The word "multinational" rarely occurs in the 1950s and

[5]See: http://kbkranten.politicalmashup.nl/#q/holocaust

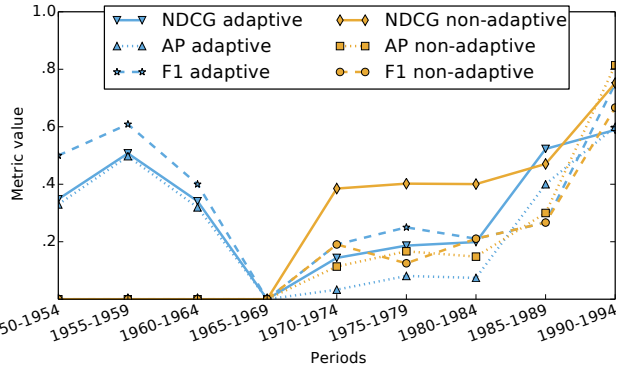

Figure 5: Comparison of results between the adaptive and non-adaptive run for the seed word "multinational." Direction is backward in time.

1960s in the Dutch digitised newspapers.[6] This is clearly reflected in Figure 5 and both the adaptive and the non-adaptive method suffer from this. Close inspection of the documents in which the word does occur in this period reveal that it is used in a political context (where it means international) rather than in a business context as later on. Importantly, the adaptive run is able to recover its drop in performance, while the non-adaptive run is unable to do so, and keeps getting zero performance.

The examples in this section clearly show the limitations of non-adaptive approach that simply follows a static set of words and the words related to them over time. If the words in the seed set simply do not exist in the period of interest (as in the "cd" example), change in meaning (the "multinational" example), or are not used throughout the entire period of interest (the "holocaust" example), a static approach can only fail.

5.1.2 *Overgeneration*

As discussed in §4.1 the evaluation set contains 5 a-historical seed term sets to check for overgenerating. In Table 3 we display the results on the a-historical subset of the ground truth seed sets, based on the same parameter settings used for Table 2.

Table 3: Results for adaptive and non-adaptive method on a-historical seed sets only

Method	F_1	NDCG	MAP
adaptive	.395	.849	.254
non-adaptive	.387	.872	.254

As we can observe from Table 3 the results between the adaptive and non-adaptive runs are comparable. None of the differences is statistically significant for $\alpha = .05$ for a two-tailed paired t-test. We conclude from these results that our adaptive method for generating shifting vocabularies over time does not overgenerate. That is, if no changes occur in a vocabulary concerning a particular topic, none are in fact picked up by the adaptive method.

5.2 Parameter analysis

To answer research question RQ2 we analyse the effect of the parameters of the generation algorithm and the aggregation algorithm. For the generation algorithm the parameters are the length of the sliding time window, minimal semantic distance ς and the method of computing degree centrality. For the aggregation algorithm we have one parameter, the vocabulary weighting function.

[6]See: http://kbkranten.politicalmashup.nl/#q/multinational

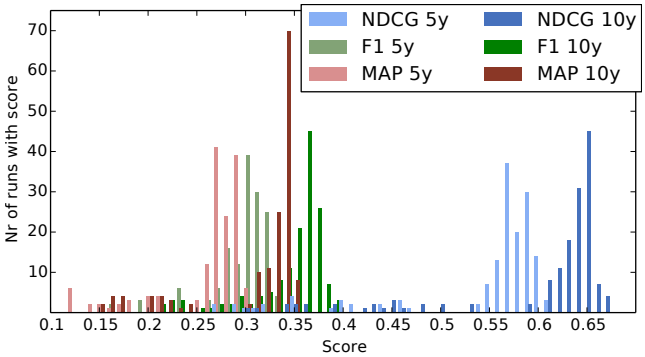

Figure 6: Comparison of results per metric, grouped by time window length. (Best viewed in color.)

Length of sliding time windows. The length of the sliding time windows affects both the adaptive method and the non-adaptive method. In Figure 6 performance of all runs — adaptive, non-adaptive and hybrid, all parameter settings — is plotted, grouped by window length. As is clear from the figure, using 10 year sliding windows yields better results in a vast majority of cases, for all metrics. In Table 4 the t-statistics and p-values are listed per metric for a paired t-test between the results per window length (the results are paired per parameter setting).

Table 4: Results of two-tailed paired t-test between the performance of all results per window length, paired by parameter setting

Metric	t-statistic	p-value
NDCG	-20.8	5.34×10^{-19}
MAP	-24.0	1.08×10^{-20}
F_1	-19.6	2.73×10^{-18}

From these findings we conclude that using a longer time window to train a semantic model yields better performance for our current task, which supports the claim made in [21] that more training data yields better semantic models. Do note, though, that, due to the adaptive nature of our task, we can not use arbitrarily long time windows, as the changes in meaning and vocabulary we are interested in might go unnoticed that way.

Minimum distance. As discussed in §3.2.1 the ς parameter controls which related words are taken into account for constructing semantic networks. In Table 5 the results across different levels of ς are displayed for all methods of generating shifting vocabularies over time, that use the generation algorithm (the non-adaptive method only uses the aggregation algorithm). The results are consistently lower than the results in Table 2, regardless of the method. This clearly indicates that a value of $\varsigma = .65$ is to be preferred for all methods, adaptive, non-adaptive or hybrid.

Table 5: Top results for different settings of minimum similarity ς, all other settings as in Table 2

Method	ς	F_1	p	r	NDCG	MAP
hybrid ($i = 1$)	.7	.367	.521	.389	.630	.332
	.6	.376	.530	.398	.637	.338
hybrid ($i = 2$)	.7	.368	.523	.385	.630	.332
	.6	.378	.534	.395	.636	.338
hybrid ($i = 3$)	.7	.372	.530	.388	.634	.335
	.6	.370	.526	.385	.632	.332
hybrid ($i = 4$)	.7	.370	.529	.381	.632	.333
	.6	.365	.521	.376	.627	.329
hybrid ($i = 5$)	.7	.366	.525	.372	.628	.331
	.6	.358	.515	.365	.622	.323
adaptive	.7	.316	.678	.241	.485	.220
	.6	.292	.442	.273	.442	.206

Degree centrality. Regarding the different ways of calculating degree centrality we observe a very consistent pattern: choosing in-degree always yields better results than choosing out-degree. The best performance with out-degree, in terms of F_1, other settings as in Table 2 is yielded by the hybrid method, with $i = 1$. It yields an F_1 of .370, NDCG of .632 and MAP of .333, all of which is lower than the scores of the best performing hybrid runs.

Putting weights on the edges consistently leads to performance superior to unweighted edges. The best performance, in terms of F_1, without weighted edges, and other settings as in Table 2 is yielded by the hybrid method with $i = 1$, which yields an F_1 of .369, NDCG of .632 and MAP of .333.

Vocabulary weighting function. In case of the non-adaptive method, not weighting the vocabularies leads to a small increase in performance: F_1 .368, NDCG .632 and MAP .333, regardless of the value for minimum similarity ς. These differences, however, are not statistically significant for $\alpha = .5$ for a two-tailed paired t-test. Furthermore, for the hybrid method, applying weighting for generating vocabularies over time nearly always yields better results when $i > 1$. These findings suggest that weighting of vocabularies is beneficial for generating shifting vocabularies over time.

5.3 Error analysis

In 9 cases of the 21, merging adaptive and non-adaptive runs for the hybrid runs led to performance that was less than the best performing of the two. In this section we will discuss three such examples. Typically, the decrease in performance was small (~1%).

Table 6: Results of the hybrid run ($i = 3$) for seed set "marxism," for the last time period (the direction in time is forward). Words occurring in the ground truth set are marked with a *.

Period	Vocabulary[7]
1990–1994	communism*, marxism*, capitalism, humanism, christianity, socialism*, imperialism, atheism, militarism *(in two different spelling variants)*

Table 6 shows the vocabulary output for the hybrid run ($i = 3$) with seed set "marxism" for the 1990–1994 period. The direction is forward in time. This means that we start with the concept of marxism in 1950 and follow it as time progresses. As we can see from the results, the adaptive run has picked up on related terms and has become too general (the concepts, though they are related, are mainstream socio-economical movements, ideologies and isms). Much more on-topic words, like, e.g., "leninism" and "stalinism," which were used in the late 1990s in the newspaper corpus are picked up by the non-adaptive run.

We see a different pattern for the run with the seed set "hydrogen bomb" in Table 7. Here, the adaptive run nearly loses track of the nuclear weapons completely, and rather focusses on missiles.[8]

The examples in this section show that the adaptive method for monitoring shifting vocabularies over time can be susceptible to

[7]The original words are in Dutch, translations by the authors

[8]The term "atomic warheads" was not annotated as correct, even though it means the same as "nuclear warhead," because it was hardly ever used, while "nuclear warhead" was used abundantly.

Table 7: Results of the hybrid run ($i = 3$) for seed set "hydrogen bomb" for the last time period (forward direction in time). Words occurring in the ground truth set are marked with a *.

Period	Vocabulary
1990–1994	launching facilities, rockets, ballistic, launching pads, nuclear warheads*, nuclear submarines, atomic warheads, nuclear payload*, multi-headed, bomber

topic drift. It can loose specificity (the "marxism" example) or it can drift in the wrong direction (the "hydrogen bomb" example). Especially in cases like this, a combination with a more conservative, non-adaptive approach is beneficial.

6. CONCLUSIONS AND FUTURE WORK

We introduced the task of ad hoc monitoring of vocabulary shifts over time. We presented several algorithms for monitoring vocabularies over time and perform systematic, intrinsic evaluation of their results. Our results show that our approach of combining an exploratory method of generating shifting vocabularies over time with a conservative approach consistently and significantly beats a baseline inspired by related research, and that it consistently performs better than the two approaches it combines.

Intrinsic evaluation of semantic methods is difficult. Constructing a manually labelled dataset as we did is costly and labour-intensive. We hope that disclosing the full evaluation set is beneficial to research in this area.

High-quality, on-topic vocabularies over time can beneficial in many cases both in IR and in digital humanities research. The vocabularies can be used as a way of exploring data, as is the underlying scenario in this paper. Furthermore however, they could be used for time-aware query expansion, where the query expansion depends on the timestamps of documents in a corpus.

Future work should focus on longer evaluation periods, e.g. a century of material. Furthermore, additional graph-based measures could be taken into account. Moreover, different types of shift in vocabulary might be discerned. Similar to how document ranking systems are tailored towards query intent, systems for monitoring shifting vocabularies over time could be optimised in terms of optimal parameter settings or choice of algorithm, depending on the type of vocabulary shift they aim to monitor.

The performance of an adaptive method for monitoring shifting vocabularies may degrade or improve over time. However, traditional evaluation metrics like NDCG or MAP are time-agnostic. Additional insights could be obtained when a time-aware evaluation metric, such as, e.g., proposed in [17] in the context of document filtering systems, would be applied to the present setting.

Acknowledgments.

This research was supported by Amsterdam Data Science, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the ESF Research Network Program ELIAS, the HPC Fund the Royal Dutch Academy of Sciences (KNAW) under the Elite Network Shifts project, the Microsoft Research Ph.D. program, the Netherlands eScience Center under project number 027.012.105, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.-013, 612.066.930, CI-14-25, SH-322-15, the Yahoo! Faculty Research and Engagement Program, and Yandex.

7. REFERENCES

[1] J. Allan. *Topic detection and tracking: event-based information organization*. Springer Science & Business Media, 2002.

[2] M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL 2014*, 2014.

[3] R. Berendsen, E. Meij, D. Odijk, M. de Rijke, and W. Weerkamp. The university of amsterdam at trec 2012. In *TREC 2012*, 2012.

[4] A. Betti and H. van den Berg. Modelling the history of ideas. *British Journal for the History of Philosophy*, 2014.

[5] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, Inc., 2009.

[6] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, 2006.

[7] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 2010.

[8] J. R. Frank, S. J. Bauer, M. Kleiman-Weiner, D. A. Roberts, N. Tripuraneni, C. Zhang, C. Ré, E. Voorhees, and I. Soboroff. Evaluating stream filtering for entity profile updates for TREC 2013. In *TREC 2013 Working Notes*. NIST, 2013.

[9] A. Gohr, A. Hinneburg, R. Schult, and M. Spiliopoulou. Topic evolution in a stream of documents. In *SDM*, 2009.

[10] J. Guldi. The history of walking and the digital turn: Stride and lounge in london, 1808–1851. *The Journal of Modern History*, 2012.

[11] K. Gulordava and M. Baroni. A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *GEMS*, 2011.

[12] D. Hall, D. Jurafsky, and C. D. Manning. Studying the history of ideas using topic models. In *EMNLP*, 2008.

[13] G. Heyer, F. Holz, and S. Teresniak. Change of topics over time-tracking topics by their change of meaning. *KDIR*, 2009.

[14] F. Holz and S. Teresniak. Towards automatic detection and tracking of topic change. In *Computational linguistics and intelligent text processing*. 2010.

[15] P. Huijnen, F. Laan, M. de Rijke, and T. Pieters. A digital humanities approach to the history of science. In *Social Informatics*. 2014.

[16] T. Kenter. Filtering documents over time for evolving topics. In *TREC 2013*, 2013.

[17] T. Kenter, K. Balog, and M. de Rijke. Evaluating document filtering systems over time. *Information Processing & Management*, 2015.

[18] Y. Kim, Y.-I. Chiu, K. Hanaki, D. Hegde, and S. Petrov. Temporal analysis of language through neural language models. In *LACSS*, 2014.

[19] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. Statistically significant detection of linguistic change. *CoRR*, 2014.

[20] J. Kuukkanen. Making sense of conceptual change. *History and theory*, 2008.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*, 2013.

[23] F. Moretti and D. Pestre. Bankspeak. *New Left Review*, 2015.

[24] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *EMNLP 2014*, 2014.

[25] A. Saha and V. Sindhwani. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In *WSDM*, 2012.

[26] K. Sparck Jones and van Rijsbergen C. Report on the need for and provision of an" ideal" information retrieval test collection. *British Library Research and Development Report 5266*, 1975.

[27] S. Wang, S. Schlobach, and M. Klein. Concept drift and how to identify it. *Web Semantics*, 2011.

[28] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *SIGKDD*, 2006.

[29] D. T. Wijaya and R. Yeniterzi. Understanding semantic change of words over centuries. In *DETECT*, 2011.