NEER: An Unsupervised Method for Named Entity Evolution Recognition[*]

Nina Tahmasebi Gerhard Gossen Nattiya Kanhabua
Helge Holzmann Thomas Risse
L3S Research Center
tahmasebi, gossen, kanhabua, holzmann, risse@L3S.de

ABSTRACT
High impact events, political changes and new technologies are reflected in our language and lead to constant evolution of terms, expressions and names. Not knowing about names used in the past for referring to a named entity can severely decrease the performance of many computational linguistic algorithms. We propose NEER, an unsupervised method for named entity evolution recognition independent of external knowledge sources. We find time periods with high likelihood of evolution. By analyzing only these time periods using a sliding window co-occurrence method we capture evolving terms in the same context. We thus avoid comparing terms from widely different periods in time and overcome a severe limitation of existing methods for named entity evolution, as shown by the high recall of 90% on the New York Times corpus. We compare several relatedness measures for filtering to improve precision. Furthermore, using machine learning with minimal supervision improves precision to 94%.

TITLE IN GERMAN: **NEER: Eine nichtüberwachte Methode zur Erkennung von Namensevolution**

ABSTRACT IN GERMAN
Wichtige Ereignisse, politische Veränderungen und neue Technologien spiegeln sich in unserer Sprache wieder und führen zu einer ständigen Evolution von Begriffen, Ausdrücken und Namen. Mangelndes Wissen über frühere Namen einer Entität kann die Leistungsfähigkeit vieler computerlinguistischer Methoden deutlich verringern. In diesem Papier präsentieren wir unsere nichtüberwachte Methode namens NEER zur Erkennung von Namensevolution, die unabhängig von externen Datenquellen arbeitet. Indem wir Zeiträume mit erhöhter Evolutionswahrscheinlichkeit mit Hilfe einer Kookkurrenzmethode basierend auf *Sliding Windows*-Verfahren untersuchen, erfassen wir evolvierende Terme im selben Kontext. Dadurch vermeiden wir es, Terme aus weit auseinander liegenden Zeiträumen zu vergleichen und umgehen damit eine schwerwiegende Beschränkung vorhandener Methoden. Dieses zeigt sich an einer gemessenen Sensitivität von 90% auf dem Korpus der New York Times. Um die Genauigkeit zu erhöhen, vergleichen wir mehrere Ähnlichkeitsmaße zur Filterung. Mit Hilfe von maschinellem Lernen mit minimaler Überwachung verbessern wir die Genauigkeit auf 94%.

KEYWORDS: Temporal Named Entity Evolution, Named Entity Changes, Machine Learning.

GERMAN KEYWORDS: Zeitliche Namensevolution, Namensänderungen, Maschinelles Lernen.

[*]This work is partly funded by the European Commission under ARCOMEM (ICT 270239)

1 Introduction

Do you remember the bright yellow Walkman, Joseph Ratzinger or Andersen Consulting? Chances are you do not, because as the world around us changes, new terms are created and old ones are forgotten. High impact events, political changes and new technologies are reflected in our language and lead to constant evolution of terms, expressions and names. Most everyday tasks, like web search, have so far relied on the good memory of users or been restricted only to the current names of entities. As the web and its content grow older than some of its users, new challenges arise for natural language tasks like information retrieval and knowledge consolidation to automatically determine relevant information, even when it is expressed using forgotten terms.

Language evolution is reflected in documents available on the web or in document archives but is not sufficiently considered by current applications analyzing them. Therefore, not knowing about different names referencing the same entity severely compromises system effectiveness. This can partially be addressed by using external knowledge sources like DBpedia. The limitation of this approach is that these resources do not cover all entities and are not able to capture ephemeral names or jargon used in everyday language or social media.

There are several kinds of language evolution, among others spelling variations, name changes and concept changes. We focus on named entity evolution, the detection of name changes, as it has a high impact, for example in information retrieval. This research field is becoming increasingly important. However, most previous work depend on the availability of external knowledge sources or assume a static context around terms and expect the names to be the only changing factor. We follow a statistical approach to eliminate the dependency on external resources and use a context based method that considers only periods with a high likelihood of name change, thereby capturing evolving names with less computational effort. This independence opens the possibility to apply the method to any corpus, including historical collections or those in different languages, and to identify undocumented named entity evolution.

The main contributions of this paper are:

- We propose the use of change periods (i.e., periods with high likelihood of name change) to capture the evolution of one name into another instead of comparing names and their contexts from vastly different periods in time.
- We propose NEER, a method for named entity evolution recognition that analyzes the context of entities during time periods of evolution. The proposed method is independent from external knowledge sources and is able to find name changes.
- We describe and compare named entity evolution filtering methods, statistical as well as machine learning based, that capture relatedness between different names used to reference the same entity at different points in time in order to increase accuracy.
- We apply NEER on a standard dataset (New York Times corpus) to identify named entity evolution and evaluate using precision and recall. We make our test set publicly available to encourage comparison of results.

The rest of the paper is organized as follows: In the next section we review related work in areas relevant to named entity evolution. In Section 3 we define terminology needed for our work. We describe our approach and highlight the limitations of previous work in Section 4 and present our methodology in Section 5. We introduce our data and test sets in Section 6 and present our experimental results. We discuss our findings in Section 7 and finally present our conclusions and future work.

2 Related Work

Previous work on automatic detection of term evolution has been very limited and mainly focused on named entity evolution. The interest has largely been from an information retrieval point of view as named entity evolution makes finding relevant documents more challenging.

Berberich et al. (2009) propose reformulating a query into terms prevalent in the past; they measure the degree of relatedness between two terms when used at different times by comparing the contexts as captured by co-occurrence statistics. This approach requires a recurrent computation each time a query is submitted as it requires a target time point for the query reformulations which reduces efficiency and scalability. The results presented in this paper are "anecdotal" (to use the words of the authors) and thus cannot be used for direct comparison. However, because of the promising results we use the same method for defining a context.

Kaluarachchi et al. (2010) propose to discover semantically identical concepts (or named entities) used in different time periods using association rule mining and associating distinct entities to events. Sentences containing a subject, a verb, objects, and nouns are targeted and the verb is interpreted as an event. Two entities are considered semantically related if their associated event is the same and the event occurs multiple times in a document archive. The temporally related term of a given named entity is used for query translation (or reformulation) and results are retrieved appropriately w.r.t. specified time criteria. They present precision and recall for very few queries and evaluate only indirectly on the basis of retrieved documents.

Kanhabua and Nørvåg (2010) define a time-based synonym as a term semantically related to a named entity at a particular time period. They extract synonyms of named entities from link anchor texts in Wikipedia articles using the full history. Unfortunately, link information, such as anchor texts, is rarely available and thus limits the method to hypertext collections. They evaluate the precision and recall of the time-based synonyms by measuring increased precision and recall in search results rather than directly evaluating the quality of the found synonyms.

3 Definitions and Terminology

Language evolution is a broad concept which can be divided into several sub-classes including spelling variations (Ernst-Gerlach and Fuhr, 2007; Hauser et al., 2007; Gotscharek et al., 2009) and word sense evolution (Tahmasebi et al., 2011).

The general class of **term to term evolution** contains terms of any part of speech that have been used to mean the same thing at different points in time. The shift between terms typically occurs over a long period of time, e.g., *cool* was previously expressed with the term *collected*. In this paper, we focus on a special case of term to term evolution namely **named entity evolution** which considers a given entity and the different lexical names for the same entity. Here, the entity is fixed while the name changes over time.

Named entity changes do not need to have any lexical overlap between two representations, for example *Joseph Ratzinger* was the birth name of *Pope Benedict XVI* and *Hillary Rodham* was *Hillary Clinton*'s maiden name. The latter is an easier case of evolution because of the overlapping first name and can be targeted using entity consolidation or linking techniques (Shen et al., 2012; Ioannou et al., 2010). However, most existing techniques do not take historic changes into account and only focus on merging concurrent representations of the same entity.

We consider a term w_i to be a single or multi-word lexical representation of an entity at time t_i. The **context** C_{w_i} is the set of all terms related to w_i at time t_i. Similar to Berberich et al. (2009)

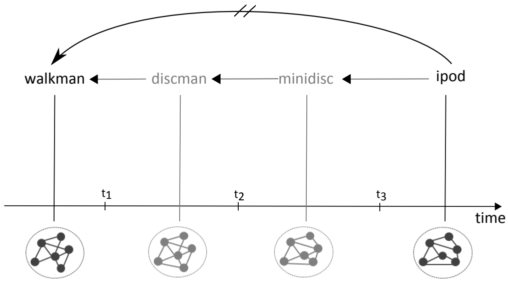


Figure 1: Existing methods detect evolution by comparing term contexts from different times. E.g., by directly comparing *walkman* to *ipod*.

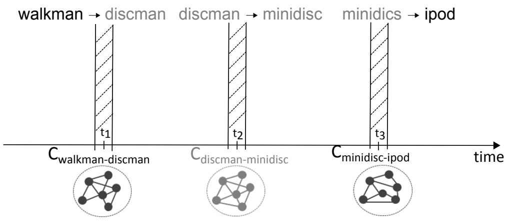


Figure 2: NEER detects by creating one context for each change period, thus eliminating the need to compare contexts.

we consider the most frequently co-occurring terms within a distance of k words as the context, however, other contexts can be used. We consider a **change period** to be a period of time in which one term evolves into another.

Co-references are expressions that refer to the same entity. In the sentence "The president said he had discussed the issue" the words *the president* and *he* refer to the same person. In this paper, we consider **temporal co-references** to be different lexical representations that have been used to reference the same concept or entity at the different periods in time.

We have two variations of temporal co-references, direct and indirect. **Direct temporal co-references** are temporal co-references that are variations of each other with some lexical overlap. **Indirect temporal co-references** are temporal co-references that lack lexical overlap on the token level. *Hillary Clinton* and *Hillary Rodham* are examples of direct temporal co-references while *Pope Benedict XVI* and *Joseph Ratzinger* are examples of indirect temporal co-references. All introduced terms will be used with and without *temporal* interchangeably.

A **temporal co-reference class** contains all direct temporal co-references for a given named entity, denoted as $\text{coref}_r\{w_1, w_2, \ldots\}$. Each temporal co-reference class is represented by a **class representative** r which is also a member of the class. For example, *Joseph Ratzinger* is the representative of the co-reference class containing the terms {*Joseph Ratzinger, Cardinal Ratzinger, Cardinal Joseph Ratzinger, . . .* }.

4 Approach

Previous work in the area can be generalized as shown in Figure 1. A word w_i (*walkman*) is mapped to its context and compared to word w_j (*ipod*) by comparing contexts. If the corresponding contexts are similar it is concluded that w_i and w_j are temporal co-references. These methods have severe drawbacks because they assume that the queried entity is the only evolving factor and that contexts stay stable over time. This is however not the case. Comparing the term *walkman* and *ipod* (an example given by Berberich et al. (2009)[1]) directly by means of contexts from the New York Times corpus ($C_{walkman}$, C_{ipod}) we find that the majority of the context terms have changed. As we can see in Table 1, the same holds for the contexts of related terms *discman, minidisc* and *mp3 player* during the years each term was introduced[2].

[1]We do not consider *walkman* and *ipod* to be co-references as they do not correspond to the same named entity. We use this example to illustrate the difficulties that arise and the different methods.

[2]The *walkman* was already introduced in 1979, so we chose the first year of the corpus' time span (1987).

$C_{walkman}$	$C_{discman}$	$C_{minidisc}$	$C_{mp3\ player}$	C_{ipod}
cassette	walkman	compact	music	apple
audio	stillvideo	disc	digital	mp3
video	sony	sony	internet	roqit
tape	portable	digital	audio	player
music	cd	cassette	player	music
sony	kodac	phillips	files	geeks
digital	video	walkman	cd	jukebox
stereo	priestly	dcc	computer	portable
earphones	digital	prerecorded	mp3	macintosh
recorders	camera	video	portable	dlink

Table 1: Five terms and their contexts in the New York Times corpus.

We find that the only overlap between $C_{walkman}$ and C_{ipod} is the term *music*. By comparing the intermediate contexts pairwise instead we find that there is a much larger overlap between the contexts. For instance, $C_{walkman}$ has a 40% overlap with $C_{discman}$ which in turn has a 30% overlap with $C_{minidisc}$. The same properties hold when we compare the 20 most frequent terms and we find that the overlap between $C_{mp3player}$ and C_{ipod} increases further. From this we deduce that comparing contexts pairwise where the contexts are closer in time is more effective than comparing two contexts far apart in time.

The same observation holds for Kaluarachchi et al. (2010). They consider nouns to have evolved into each other if they point to the same event (represented by a verb) at different points in time. Over long periods of time also the verb undergoes evolution and hence the method is limited only to those terms where the corresponding event has not changed over time.

In this paper we make use of the typical characteristics of named entity evolution. Unlike with other types of evolution, such as word sense evolution, named entity changes typically occur during a short time span. There are few concept shifts where the term slowly changes, instead name changes occur due to special events like being elected pope, getting married or merging/splitting a company. If the named entity is of general interest, these name changes will also be announced to the public repeatedly during the change period with sentences like “The day after Cardinal Joseph Ratzinger became Pope Benedict XVI”[3].

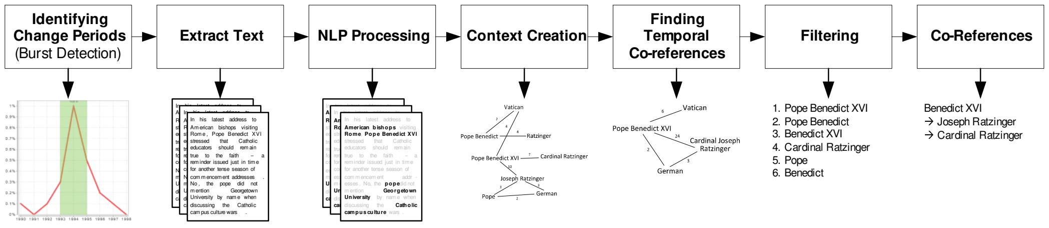
By first identifying candidate change periods and then creating a context around a term, we believe that we can capture both the old and the new co-reference in the same context. We thus eliminate the risk of comparing contexts that are vastly different. Figure 2 illustrates our method. By identifying change periods t_1, t_2, t_3 we can create contexts around a term which contain both co-references and do not have to compare largely different contexts like those of *walkman* and *ipod*.

5 The NEER Algorithm

To find temporal co-references we use the pipeline depicted in Figure 3. We start by detecting change periods for a query term over the entire collection. We make use of the identified change periods to find the subsets in which we look for evolution. We extract single and multi-word nouns and find named entities mentioned in the text.

We create contexts around extracted terms by applying co-occurrence analysis and use the context and the extracted terms to find direct co-references. Finally we apply frequency analysis as well as machine learning to identify direct and indirect co-references and filter out noise.

[3] The New York Times, April 21, 2005.



In his latest address to American bishops visiting Rome, Pope Benedict XVI stressed that Catholic educators should remain true to the faith – a reminder issued just in time for another tense season of commencement addresses. No, the pope did not mention Georgetown University by name when discussing the Catholic campus culture wars.

Figure 3: Pipeline used to detect temporal co-references.

5.1 Identifying Change Periods

Named entity changes are typically associated with significant events concerning the entities which lead to increased attention. We use this property to pinpoint change periods and detect those using a **burst detection algorithm**.

We use the Kleinberg algorithm (Kleinberg, 2003) to find bursts from the entire document collection $\mathbf{D}$. The algorithm models the frequency of documents $\mathbf{D}_w$ containing term w using a series of probability distributions. Each distribution represents an increasing degree of burstiness. A set of states indicates which distribution is active. By assigning a cost to state transitions the algorithm ensures that an optimal state sequence creates bursts that end only if they are followed by a sufficiently large period of lower activity. This avoids splitting bursts for example around weekends when the number of articles drops.

We detect bursts related to an entity by retrieving all documents in the corpus containing the query term, grouping them into monthly bins and running the burst detection on the relative frequency of the documents in each bin. Each resulting burst corresponds to a significant event involving the entity. However, these bursts do not necessarily correspond to a name change. By choosing the `topB` strongest bursts we expect to find a subset of bursts which also capture change periods. We denote each change period p_i for $i = 1, \ldots, \mathtt{topB}$.

5.2 Creating Contexts

After identifying change periods p_i for an entity w we create a context for each period by extracting all documents $\mathbf{D}_{w_i}$ that mention the entity or any part of it and are published in the year corresponding to p_i. We lemmatize the text and extract nouns, noun phrases and named entities. We use noun phrases to capture more information and create richer contexts around entities. All extracted terms are added to a **dictionary** and used for creating a co-occurrence graph. The co-occurrence graph is an undirected weighted graph which links two dictionary terms if and only if they are present in $\mathbf{D}_{w_i}$ within k terms of each other. The weight of each link is the frequency with which the two terms co-occur in $\mathbf{D}_{w_i}$. The context of entity w is considered as all terms co-occurring with w. The context of a co-reference class is considered to be all terms co-occurring with any of the terms in the co-reference class.

5.3 Finding Temporal Co-references

To find *direct co-reference classes* we need to consolidate the extracted terms by recognizing all variants of each term. As an initial step each term from the dictionary with a frequency above `minFr` is placed in its own co-reference class where the term acts as the representative as well as the only co-reference: $\text{coref}_{\text{Benedict}}\{\text{Benedict}\}$.

Merging: The procedure for merging terms and co-reference classes is shared between all three rules described below; each co-reference class is represented by the term with the highest frequency. A frequency is stored in the co-reference class for the representative r as the sum frequency of all terms in the class. If two co-reference classes have the same representative, they are merged into one. Each co-reference class carries with it all co-occurrences that belong to any of the terms in the co-reference class. These are considered as the context C_r. When terms are merged, the context is updated accordingly.

Next we will describe the main rules used for finding all direct co-reference classes. In the initial iteration the first rule works on the dictionary terms and populates an index with co-reference representatives. In the second and all subsequent iterations the first rule makes use of the terms in the index. This index is passed through all the rules. The rules are iterated until there are no more terms in the index that can be merged.

1. **Prefix/suffix rule:** This rule creates co-reference classes by merging dictionary terms that differ only by a prefix or suffix. For example, the co-reference classes of *Pope Benedict* and *Benedict* as well as *Pope* and *Pope Benedict* are merged. In both cases the resulting co-reference class has *Pope Benedict* as representative and these co-reference classes are therefore merged: $\text{coref}_{\text{Pope Benedict}}\{\textit{Pope, Pope Benedict, Benedict}\}$.
2. **Sub-term rule:** This rule merges classes that are represented by terms that can be considered sub-terms. For a term to be a sub-term of another we require the longer term to contain all terms from the shorter term in the correct order. For example, the co-reference class represented by *Cardinal Joseph Ratzinger* and *Cardinal Ratzinger* are merged.
3. **Prolong rule:** The third rule is used to create longer terms than might be found in the dictionary. It merges two representatives from the index into one longer term if the terms have an overlapping part and there exists a co-occurrence between the remaining terms. E.g., *Pope John Paul* and *John Paul II* are merged if there is a co-occurrence (*Pope John Paul* , *II*) or (*Pope* , *John Paul II*); the representative of the merged co-reference class is *Pope John Paul II*. The third rule also merges terms that differ due to plural of the prefixes assuming that the prefix is not considered a stopword. E.g., *Senator Barack Obama* and *Senators Barack Obama* are merged but *Mr Obama* and *Mrs Obama* are not.

Final merging When the terms in the index cannot be merged further, a final round of merging takes place. In this round we apply a *soft* sub-term rule where we drop the requirement that the terms should be in the same order but require them to be similar in frequency. This way terms like *Illinois Democrat* and *Democrat of Illinois* are merged.

Consolidation When all terms are merged we create a mapping from each term to the co-reference class representative that has the highest frequency. Using this map we consolidate all terms in the context of each class.

An example of is shown in Figure 4 (original context in 4a). Using the three rules we find

$$\begin{aligned}
\text{coref}_{\textit{Cardinal Joseph Ratzinger}}&\{\textit{Cardinal Joseph Ratzinger, Joseph Ratzinger, Ratzinger}\}\\
\text{coref}_{\textit{Pope Benedict XVI}}&\{\textit{Pope Benedict XVI, Pope Benedict,Pope}\}\\
\text{coref}_{Vatican}&\{Vatican\}\\
\text{coref}_{German}&\{German\}.
\end{aligned}$$

Next a mapping is created: [*Joseph Ratzinger* → *Cardinal Joseph Ratzinger*, *Ratzinger* → *Cardinal*

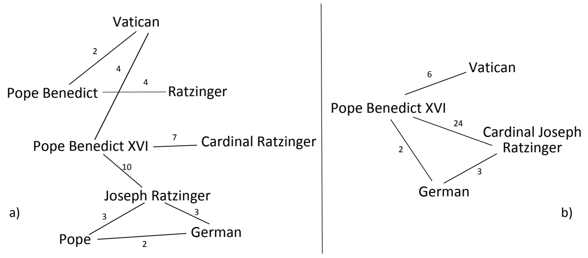


Figure 4: a) Example graph after creating context. b) After consolidating and merging all direct co-references.

Joseph Ratzinger, Pope Benedict → *Pope Benedict XVI, Pope* → *Pope Benedict XVI*]. Additionally, all co-reference class representatives map to themselves. Then each term in the co-reference class context is consolidated and replaced using the map. If two co-occurrences share a term they are merged into one and the frequency of the co-occurrence is updated as shown in Figure 4b.

Ranking The term frequencies and the merging steps offer a natural ranking of co-references. When two terms are merged like *Pope Benedict* and *Pope Benedict XVI* we update the frequency of the class representative by summing up the frequencies. During merging all co-occurrences are updated with the sum of the frequencies of all participating terms. In Figure 4b we see that the co-occurrence frequencies of (*Vatican* , *Pope Benedict XVI*) is 6 because the frequency of (*Vatican* , *Pope Benedict*) is 2 and (*Vatican* , *Pope Benedict XVI*) is 4. The term frequencies and co-occurrence frequencies are stored in each co-reference class. The frequency of *Pope Benedict* and *Pope Benedict XVI* is much higher than that of *Benedict XVI* and *Eggs Benedict* and during consolidation the term *Benedict* is replaced with *Pope Benedict XVI* rather than *Eggs Benedict*.

Indirect Co-references Indirect co-references are found implicitly by means of the direct co-references. After consolidation, all terms in the context of a co-reference class are considered candidate indirect co-references. These are a mix between true indirect co-references, highly related co-occurrence phrases as well as noise. The quality of the indirect co-references is dependent on the named entity extraction, co-occurrence graph creation and filtering of the co-occurrence graph. The choice of including single token terms in addition to multi-token terms has a high influence on the quality of the resulting co-occurrences.

If NEER does not find any co-references for a term, all direct co-occurrences from the co-occurrence graphs (derived from the union of the change periods) are returned instead.

5.4 Filtering Temporal Co-references

To remove noise and identify the true direct and indirect co-references we make use of the term frequencies as well as the document frequencies for the filtering. We start by describing similarity measures between terms and continue with the filtering techniques.

Similarity measures To keep true co-references we need to measure the temporal relatedness of terms. Unlike previous works that take temporal features into account it is not sufficient to consider relatedness over the entire time span of a collection. In Radinsky et al. (2011) time series of terms are used to capture the relatedness of terms like *war* and *peace* or *stock* and *oil*. These terms are considered related because they have similar frequencies over time.

For temporal co-references, capturing the overall relatedness is not sufficient. Both direct and indirect co-references can be related only for a certain period in time and then lose their relation. To give an example: Both *Senator Clinton* as well as *Hillary Clinton* have been used to refer to the same person at different periods in time. As the latter name was only valid for a certain period in time measuring the relatedness between *Hillary Clinton* and *Senator Clinton* using global term frequencies (i.e. term frequency over the entire corpus) will not yield the correct results. However, global measures can help to find direct co-references such as *Barack Obama* and *Barack Hussein Obama*.

To fully capture temporal co-references we need, in addition to global relatedness measures, a relatedness measure that captures how related terms are during the time periods where they can be related at all. To this end we allow a relatedness measure to consider only periods where both terms occur. In all cases we use the normalized frequencies.

We consider four relatedness measures: (1) Pearson's Correlation ($corr$) (Weisstein, 2012a), (2) Covariance (cov) (Weisstein, 2012b), (3) Rank correlation (rc) and (4) Normalized rank correlation (nrc).

The two first measures are standard relatedness measures where $corr$ measures linear dependence between random variables while cov measures correlation between two random variables. The two last measures are rank correlation measures and inspired by the Kendall's tau coefficient that considers the number of pairwise disagreements between two lists. Our rank correlation coefficient counts an agreement between the frequencies of two terms for each time period where both terms experience an increase or decrease in frequency without taking into consideration the absolute values. The rank correlation is normalized by the total number of time periods. The normalized rank correlation considers the same agreements but is normalized with the total number of time periods where both terms have a non-zero term frequency.

Filtering Co-references using Pearson's Correlation The first filtering makes use of the correlation measure to determine which co-references are related to the query term and filter out the rest. This measure is used in (Radinsky et al., 2011) to measure similarity between terms and serves as a comparison for our filtering mechanisms. We keep a co-reference if its correlation to the query term exceeds the threshold cor_{min}. An increase in the filtering threshold would lead to the same or decreased recall while the precision could be affected either way. A decrease in the threshold would lead to a lower precision. Therefore a low threshold is sufficient to get an upper bound of the recall while maintaining precision.

Filtering Co-references using Document Frequency The second filtering is based on the document frequencies (df) of co-references. We filter out all co-references that differ largely in document frequency from the document frequency of the query term. The filtering depends on the document frequency of the most frequent term in the dictionary corresponding to a change period ($\text{df}_{\texttt{max}}$) and a *scaling factor* (sc). We filter out all co-references that have a document frequency $\text{df} \geqslant \text{df}_{\texttt{max}} \cdot \text{sc}$, i.e., which are frequently used in different contexts.

Filtering Co-references using Machine Learning Our third and final filtering method is based on machine learning. We use a random forest classifier (Breiman, 2001) consisting of a combination of decision trees where features are randomly extracted to build each decision tree. In total ten trees with three features each are constructed. We choose features from the similarity measures presented above. That means for each term–co-reference pair (t, c) found by NEER we calculate the $corr$, cov, rc and nrc measures. We also use the average of all four measures as a fifth feature. Finally we classify the pair as either 1 for c being a correct co-reference of t or 0 otherwise.

6 Experiments

The aim of our experiments was to measure how well NEER can detect names used during different time periods to refer to the same entity. We did this by (1) investigating how well burst detection can be used to capture change periods, and (2) measuring precision and recall of the co-references found using NEER. Each experiment in (2) was performed using two settings: (a) The first made use of the known change periods (denoted **known periods**), (b) the second used the detected bursts (denoted **found periods**). We used the known change periods to measure how well the method works assuming that we can find the correct change periods.

As there are no available baselines to compare our methods to, we defined our own baseline and named it **co-occurrence**. This considers all terms that co-occur with the queried named entity within a sliding window for all change periods in (a) and (b). This provided a baseline that shows what can be achieved with minimal computational effort.

We considered $precision = \frac{\text{\# correctly captured co-references}}{\text{\# all captured co-references}}$ and $recall = \frac{\text{\# captured co-references}}{\text{\# known name changes}}$ for our evaluation. For a term we required all direct co-references and at least one indirect co-reference for each name change to achieve full recall. That means that for *Joseph Ratzinger* we required all direct forms {*Cardinal Joseph Ratzinger, Cardinal Ratzinger*} but only one of the indirect {*Pope Benedict XVI, Pope Benedict*} to achieve a recall of 100%.

6.1 Experimental Setup

Dataset and Test Set For our experiments we used the New York Times Annotated Corpus (NYTimes). The dataset contains around 1.8 million articles published between 1987 and 2007.

We devised a test set of named entities, based on Kanhabua and Nørvåg (2010), with direct as well as indirect co-references and divided them into three categories: People, Locations and Companies. We identified all relevant name changes and the year in which they occur. Each co-reference pair was verified using three judges and kept if at least two judges agreed. If the change occurs in January of a year also the previous year was added. Change periods are available in the released test set. We mirrored all the entities so that *Pope Benedict* → *Cardinal Ratzinger* and *Cardinal Ratzinger* → *Pope Benedict* both exist as separate entries.

The final test set was devised by keeping all terms that (1) exist in the NYTimes, (2) have a change period in the NYTimes time span and (3) occur at least 5 times in at least one change period. The dataset is available in Tahmasebi et al. (2012). We started with 43 distinct entities and 644 co-references. After filtering there were 16 distinct entities corresponding to 33 terms and 86 co-references (44 indirect and 42 direct).

Setup We used the NYTimes API to extract documents from the NYTimes corpus. To extract terms we used Lingua English Tagger (Coburn, 2008) for finding single and multi-token nouns and the Stanford Named Entity Recognizer (NER, Finkel et al., 2005) to extract named entities. NERs typically consider names but not the role as a part of the name. For example *Barack Obama* is extracted but not *Senator Barack Obama*. Therefore we used the Lingua tagger which recognizes also terms like *Senator Barack Obama*. Named entities recognized by both methods

	Found periods			Known periods		
Method	**Precision**	**Recall**	**avr # co-ref**	**Precision**	**Recall**	**avr #co-ref**
co-occurrence	8%	51%	120	20%	59%	16
NEER	8%	90%	128	13%	89%	64
NEER + Corr	20 %	61%	107	17%	74%	43
NEER + DF	33%	86%	28	50%	81%	10
NEER + ML	91 %	81%	6	94%	92%	5

Table 2: Precision and recall for the baseline and the different filtering techniques.

are counted twice and thus receive a higher frequency. This procedure helps to choose good representatives for the co-reference classes.

In order to increase precision we filtered out infrequent terms. During graph creation we required a term to occur at least three times in the collection used for creating the graphs. If the most common terms in the dictionary occured more than 800 times, we required at least five occurrences. For finding direct co-references we required that each term occur at least five times. However, if the most common term in the dictionary occured more than 3000 times we increased the threshold to 10 occurrences. We also filtered out terms containing lowercase tokens. For this reason the term *Union of Myanmar* could not be found by the system.

For the relatedness calculations we used normalized term frequencies that are calculated as the fraction of term occurrences in all documents published per month divided by the total number of tokens in these documents.

To find the bursts we used the Java implementation from CIShell (Alencar, 2012) with 3 burst states, a transition probability γ of 0.8 and a density of 1.9. Using these parameters we detected on average 3.2 bursts for each term in our test set.

6.2 Results

Burst Detection We approximated change periods using burst detection. Considering all found bursts for an entity we were able to capture 73% of all change periods. This indicates that burst detection works well for capturing burst periods but that there is room for improvement and parameter tuning. To reduce complexity and false positives, we limited the number of bursts to $\texttt{topB} = 6$. Using the $\texttt{topB}$ bursts we were able to capture 66% of all change periods.

If a name is ambiguous the bursts are less suitable for capturing correct change periods as the burst detection algorithm cannot distinguish between entities. This is the case for *George Bush* where the top bursts are 1988, 1989 and 1990 and correspond to *George Bush Senior*. *George W Bush* is not ambiguous and the found bursts are 1999, 2000 and 2001.

The results for the named entity evolution detection presented next are summarized in Table 2.

Baseline – Co-occurrence For comparison we chose a baseline consisting of all terms that co-occur with the query term in the datasets corresponding to the known and found burst periods. This naive baseline serves as a lower bound on recall. We used precision and recall for direct and indirect co-references found by our method and the corresponding entries from the test set. In Table 2 we find that the recall for the baseline is 60% using known periods and 51% using found periods. When considering the co-occurrences for the query term very few or no direct co-references were found for the terms. Instead most indirect co-references were

Barack Obama	**Vladimir Putin**	**Sean Combs**
Senator	President-elect Vladimir V Putin	Puffy
State Senator Barack Obama	Minister Vladimir Putin	Sean John
Senator-elect Barack Obama	Acting President Vladimir V Putin	Diddy
Senator Barack Obama	President Vladimir V Putin	Sean Combs Ruiz
Illinois Democrat	Vladimirovich	Sean John Combs

Table 3: Terms and their top temporal co-references. Gray cells are considered incorrect.

found. The precision differs largely between known and found periods; for the latter precision is surprisingly high with 20% compared to the found bursts. This shows that the known periods help to find better co-references without introducing too much noise.

To mimick other methods where the user chooses a target time, we randomly chose three years per query term (corresponding to the average number of bursts per term) and created co-occurrence graphs for these years. We repeated the experiment three times and got an average recall of 36% which is statically significantly lower than the recall for known burst showing the power of using change periods for finding temporal co-references. As a comparison, a baseline method that chooses all terms that have lexical overlaps with the query term, can maximally achieve a recall of 49% (= 42/86) because no indirect co-references can be found.

NEER In this experiment we kept all co-references found by NEER without any filtering. This experiment provides an upper bound on the recall for the subsequent experiments. We found that for known periods as well as for found periods recall is high with 89% and 90% respectively. Only very few true co-references were missed and we found at least one correct co-reference for all but two terms. The precision is much lower for known bursts in comparison to the baseline which is a consequence of the higher average number of co-references found. However the recall is statistically significantly higher. The precision of the found bursts is comparable to the baseline and again the recall is significantly higher.

Out of 22 terms with indirect co-references our method was able to find at least one indirect co-reference for 21 terms for both known burst and found bursts. For found bursts no indirect co-reference could be found for *Airtran* because no bursts could be detected. For known bursts no indirect co-references could be found for *Andersen Consulting*.

Some sample queries and their five most frequent direct and indirect co-references for known bursts are shown in Table 3. As we can see the results contain co-references of high quality. For *Vladimir Putin* NEER found four roles *President-elect, Minister, Acting President* and *President*. For *Sean Combs* all but one of his names are present in the top five co-references, missing is only *Puff Daddy* which appears on a lower rank. *Sean Combs Ruiz* is an error caused by the term extraction. The term *Ruiz* is a name of a movie character that was played by Sean Combs in 2001. For *Barack Obama* we found the term *Senator*.

Considering names with only a single token typically decreases the precision for people because it increase the number of co-occurrences with first names. However, for companies and people it is necessary to keep single token names as otherwise many names would be missed. To further improve results an extension to NEER could classify names into different categories. The extended NEER can then keep or discard single token names accordingly as well as allow different name patterns such as names with non-capital tokens (e.g., *Union of Myanmar*).

NEER + Correlation filtering Using correlation as a filtering, with $\text{cor}_{\text{min}} = 0.4$, precision increased over the NEER results while recall decreased. For both known and found periods the decrease in recall corresponds to a statistically significant decrease. The recall is higher than that of the baselines but is not competitive to the NEER results and shows that the correlation is not an appropriate similarity measure for temporal co-references.

NEER + Document Frequency filtering In this experiment co-references were removed if they occured in more documents than the query term times a scaling factor (sc), as described in Section 5.4. The filtering provides a good recall for both found and known periods. The decrease in recall compared to the NEER results is not statistically significant for either found nor known periods. With regards to precision both found and known periods show the most competitive performance compared to the baseline and the NEER results.

We used $\text{sc} = 10$ for $\text{df}_{\texttt{max}} \leqslant 300$, 5 for $\text{df}_{\texttt{max}} \leqslant 800$ and 3 for $\text{df}_{\texttt{max}} > 800$. These filtering thresholds as well as the scaling factors were found empirically. Learning these could improve the results for document filtering further.

NEER + Machine Learning We showed that unsupervised filtering can perform well for filtering out erroneous co-references found by NEER. In this experiment we investigated if the results could be further improved by using a limited amount of supervision for training a classifier. We used WEKA (Hall et al., 2009) and the random forest classifier. We trained our classifier on the dataset and used 15 fold stratified cross fold validation to determine precision and recall. We classified each term–co-reference pair produced by NEER. For known burst we got in total 2963 instances where 170 were correct co-references (we accepted combinations of correct names, e.g., *Sean Diddy Combs*). Using the classifier we were able to achieve a 94% precision and only 11 false co-references were classified as correct. The recall of the filter is 92%, however, it was only applied to the output of the NEER. For the known bursts this corresponds to a true recall of $92\% \cdot 89\% = 82\%$. This shows the true potential of the machine learning approach and the features chosen for the classification. Assuming that the NEER results can be improved further the classification is a powerful tool.

For the found bursts there were 7048 instances with 204 correct co-references. The precision of 92% is comparable to that of the known bursts and only 16 false co-references were classified as true co-references. The recall is lower with 81% and is likely a consequence of the ratio between the two classes with 204 vs 6844 instances. Adding learning instances could help boost the results, e.g., using all terms from the test set could further improve the recall.

7 Discussion

Our experiments show that we are able to find temporal co-references with high recall without depending on external knowledge sources. We found that, even though not all change periods could be found using burst detection (recall 66%), we still managed to get a recall that is comparable to the high recall for the known (correct) change periods. Because every change period captures two co-references, it is possible to capture more co-references than the number of found change periods suggests.

There can be several reasons for a change period not to occur as a burst. In some cases the name change is discussed before the change takes place and thus there is a discrepancy between the found burst and the 'true' change period. It is also possible for a name change to correspond to a smaller increase in frequency than other events (possibly such leading up to and causing

Accenture	Comcast	Czech Republic	Myanmar
Accenture Match Play Championship	Comcast Corporation	Hungary	Burma
Andersen Consulting	AT&T Comcast	Slovakia	

Table 4: Top co-references from the categories *Location* and *Company* after document frequency filtering for known bursts. For *Myanmar* only *Burma* remains after filtering.

the name change). Future work is to learn thresholds to find bursts that correspond to change periods or find other methods better suited for change period detection.

We found that co-references cannot be detected in a symmetric way: Finding w_i as a co-reference for w_j does not imply that we can find w_j as a co-reference for w_i. E.g., NEER found *Slovakia* and *Czech Republic* as co-references for *Czechoslovakia*. However, for *Czech Republic* NEER could not find *Czechoslovakia* (using found or known bursts).

Our experiments show that co-references found for terms from the category *People* have a good accuracy among the top co-references also without filtering. However, for the category *Locations* and *Companies* filtering is needed for achieving high accuracy among the top results. Some examples for companies and locations can be found in Table 4. For *Accenture*, *Czech Republic* and *Myanmar* we found an indirect co-reference among the top two terms.

By making use of terms from the dataset we ensure that all found temporal co-references can be used for information retrieval on the dataset. The results of Kanhabua and Nørvåg (2010) contain co-references of high quality like *Senator Barack H. Obama Jr*, but many of these do not appear in the NYTimes and thus cannot be used to retrieve documents. By not relying on external resources we enable a robust method that can be applied on any corpus and finds ephemeral co-references like *President-elect George Bush* or *Senator-elect Barack Obama*. NEER can also be applied to heterogeneous data such as long-term archives as well as web data and can mix content from several sources.

Conclusions and Future Work

In this paper we presented NEER, an unsupervised approach for named entity evolution recognition that overcomes limitations of existing approaches and does not depend on external knowledge sources. We made use of change periods to create term contexts to capture co-references in the same context, thereby avoiding to compare term contexts from vastly different periods in time. Burst detection was used to detect change periods and captured 66% of all change periods. Because term contexts created in change periods capture more than one co-reference, 90% of all name changes were found. We used frequency analysis to find direct and indirect co-references by filtering on document frequency as well as using machine learning to classify correct co-references. Using a random forest classifier we achieved a precision of 94% on known periods and 91% on found periods. All name changes used in our test set are released to encourage further research in this area (Tahmasebi et al., 2012).

In this paper we focused on change periods for one term and searched for temporal co-references. This means that for terms with indirect co-references, we can find at most one change at a time. In future work we will focus on automatically creating chains of evolution to handle terms with many changes and to associate validity period to each co-references, e.g.,

$\textit{ipod} \xrightarrow[2012-2001]{} \textit{mp3 player} \xrightarrow[2001-1996]{} \textit{minidisc} \xrightarrow[1996-1992]{} \textit{discman} \xrightarrow[1984-1979]{} \textit{walkman}.$

References

Alencar, A. (2012). CIShell Burst Detection. Available at http://wiki.cns.iu.edu/display/CISHELL/Burst+Detection.

Berberich, K., Bedathur, S. J., Sozio, M., and Weikum, G. (2009). Bridging the terminology gap in web archive search. In *WebDB*.

Breiman, L. (2001). Random forests. In *Machine Learning*, pages 5–32.

Coburn, A. (2008). Lingua::EN::Tagger – Part-of-speech tagger for English natural language processing. http://search.cpan.org/~acoburn/Lingua-EN-Tagger-0.15/Tagger.pm.

Ernst-Gerlach, A. and Fuhr, N. (2007). Retrieval in text collections with historic spelling using linguistic and spelling variants. In *JCDL*, pages 333–341.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*, pages 363–370.

Gotscharek, A., Neumann, A., Reffle, U., Ringlstetter, C., and Schulz, K. U. (2009). Enabling information retrieval on historical document collections: the role of matching procedures and special lexica. In *Workshop on Analytics for Noisy Unstructured Text Data*, AND '09, pages 69–76.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Hauser, A., Heller, M., Leiss, E., Schulz, K. U., and Wanzeck, C. (2007). Information access to historical documents from the early new high german period. In *Digital Historical Corpora – Architecture, Annotation, and Retrieval*, number 06491 in Dagstuhl Seminar Proceedings.

Ioannou, E., Nejdl, W., Niederée, C., and Velegrakis, Y. (2010). On-the-fly entity-aware query processing in the presence of linkage. *PVLDB*, 3(1):429–438.

Kaluarachchi, A. C., Varde, A. S., Bedathur, S. J., Weikum, G., Peng, J., and Feldman, A. (2010). Incorporating terminology evolution for query translation in text retrieval with association rules. In *CIKM*, pages 1789–1792.

Kanhabua, N. and Nørvåg, K. (2010). Exploiting time-based synonyms in searching document archives. In *JCDL*, pages 79–88.

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.

Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *WWW*, pages 337–346.

Shen, W., Wang, J., Luo, P., and Wang, M. (2012). Linden: linking named entities with knowledge base via semantic knowledge. In *WWW*, pages 449–458.

Tahmasebi, N., Gossen, G., Kanhabua, N., Holzmann, H., and Risse, T. (2012). Named entitiy evolution dataset. Available online at http://l3s.de/neer-dataset/.

Tahmasebi, N., Risse, T., and Dietze, S. (2011). Towards automatic language evolution tracking, a study on word sense tracking. In *Workshop on Knowledge Evolution and Ontology Dynamics (EvoDyn 2011)*.

Weisstein, E. W. (2012a). Correlation coefficient. `http://mathworld.wolfram.com/CorrelationCoefficient.html`.

Weisstein, E. W. (2012b). Covariance. `http://mathworld.wolfram.com/Covariance.html`.