Automatic Detection of Terminology Evolution*

Nina Tahmasebi

L3S Research Center, Appelstr. 4, DE-30167 Hannover, Germany
tahmasebi@l3s.de

Abstract. As archives contain documents that span over a long period
of time, the language used to create these documents and the language
used for querying the archive can differ. This difference is due to evo-
lution in both terminology and semantics and will cause a significant
number of relevant documents being omitted. A static solution is to use
query expansion based on explicit knowledge banks such as thesauri or
ontologies. However as we are able to archive resources with more varied
terminology, it will be infeasible to use only explicit knowledge for this
purpose. There exist only few or no thesauri covering very domain spe-
cific terminologies or slang as used in blogs etc. In this Ph.D. thesis we
focus on automatically detecting terminology evolution in a completely
unsupervised manner as described in this technical paper.

1 Introduction

Finding relevant information in documents or resources that are older than one's
lifespan is difficult if one has no prior knowledge about the language of that time.
As language evolves, past language is forgotten and new terminology is adapted
reflecting on written text produced. As such texts are stored in archives; the
evolving language becomes present in the archive. When an archive user with-
out prior knowledge about these changes queries the archive, the results will be
limited to the user's terminology. This leads to a semantic gap between the ter-
minology used when querying and the terminology used for creating documents
that are stored in the archive. While we previously could visit a library and ask
for expert advice from a librarian, the Web era now provides us with more chal-
lenges. Firstly the amount of information on the Web vastly succeeds the amount
of information found in any one library; secondly new domains are established
and changing at a faster rate. There are no longer any domain experts to help
us query the knowledge banks, which results in an increasing need of automatic
ways to find terminology evolution. As an example of terminology evolution in
an archive, let us take the Russian city of St. Petersburg. In 1703 it was named
St. Pieter Burh and shortly after it was renamed to St. Petersburg. It kept the
name until 1914 when it changed to Petrograd and then in 1921, to Leningrad.
The name stayed Leningrad until 1991 when it changed back to St. Petersburg.
When having no prior knowledge of these changes, a query of the current name

---

"St. Petersburg" in The Times[1] spanning from 1785 to 1985 returns less than 80% of the relevant documents. This goes to show that terminology evolution does affect the result space of search in long term archives. Another type of terminology change is represented by the term "rock" which added the meaning "music" to its previous "stone". Language can also change due to political correctness; "fireman" is now more often referred to as "fire fighter" due to an increasing amount of women in the profession. For a classification see [17].

The main contribution of this Ph.D. thesis is to overcome these difficulties by (1) proposing a model for describing terminology evolution. Furthermore we will (2) propose a completely automatic method for detecting terminology evolution which can be divided into (a) finding evolution of word senses and (b) from this deriving at evolution of terms. We will also (3) develop methods for evaluating the outcome. Task (1) is expeded to finsih in 2009, task (2) in 2011. The final deadline for this thesis is 2012.

2   Related Work

The act of automatically detecting terminology evolution given a corpus can be divided into two subtasks. The first one is to automatically determine, from a large digital corpus, the senses of terms. This task is generally referred to as Word Sense Discrimination. The second task is to automatically detect evolution. To our knowledge little previous work has been done directly in this topic and thus we mostly investigate state of the art in related fields.

2.1   Word Sense Discrimination

In this thesis we review Word Sense Discrimination (WSD) techniques, which is the formal term for techniques used to find word senses given a collection of texts. These techniques can be divided into two major groups, supervised and unsupervised. Due to the vast amounts of data found on the Web and in Web Archives, we will be focusing on unsupervised techniques. Most WSD techniques are based upon clustering which can be divided into hard and soft clustering algorithms. In hard clustering an element can only appear in one cluster, while soft clustering allows each element to appear in several. Due to the polysemous property of words, soft clustering is most appropriate for Word Sense Discrimination.

In Schütze [15] it is said that the basic idea of WSD is to induce senses from contextual similarities. Using the Buckshot algorithm Schütze clusters second order co-occurrence vectors from a training set into coherent clusters representing word senses. The centroids of each cluster are used to label ambiguous words from the test set. An alternative approach is proposed in [7] where a word similarity measure based on [6] is presented. By evaluating an automatically created thesaurus using this similarity measure, it is shown that the measure is appropriate for word senses. Based on this a clustering algorithm called Clustering

---

[1] http://archive.timesonline.co.uk/tol/archive/

By Committee (CBC) is presented in [14]. It aims to form as many committees as possible under the condition that the newly formed committee is not similar to an existing committee. The paper also proposes a method for evaluating the output of a WSD algorithm to WordNet [10], which has since been widely used [2,3,5]. CBC is shown to outperform other known algorithms such as Buckshot and K-means, in both recall and precision. A third category of WSD techniques is presented in [4]. Here a network of co-occurring words is built from the collection. A clustering is made based on the clustering coefficient of each node, also referred to as curvature. A more thorough investigation of the curvature measure and the curvature clustering algorithm is made in [3]. An evaluation of the curvature algorithm is made on the BNC corpus using the methods proposed in [14]. A higher precision than the one reported for CBC is found and it is noted that the high performance of curvature comes at the expense of low coverage.

2.2   Detecting Evolution

Analysis of communities and their temporal evolution in dynamic networks has been a well studied field in recent years. A community can be modeled as a graph where each node represents an individual and each edge represent interaction among individuals. When it comes to detecting evolution, the more traditional approach has been to first detect community structure for each time slice and then compare these to determine correspondence [8]. These methods can be argued to introduce dramatic evolutions in short periods of time and can hence be less appropriate to noisy data. Representing the traditional approach a framework called Monic [16] is proposed for modeling and tracking cluster evolutions. The disadvantages of the method are that the algorithm assumes a hard clustering and that each cluster is considered a set of elements without respect to the links between the elements of the cluster. In a network of lexical co-occurrences this can be valuable since the connections between terms give useful information to the sense being presented. In [12] a way to detect evolution is presented which also takes in to the account the edge structure among cluster members.

   In contrast to the traditional approach [8] proposes a framework called FacetNet. FacetNet discovers community structure at a given time step t which is determined both by the observed data at t and by the historic community pattern. FacetNet is unlikely to discover community structure that introduces dramatic evolutions in a very short time period. Depending on the characteristics of the word graph derived from our collection it might be a suitable approach to filter out noise. An alternative method of finding evolutions in networks can be inspired by [11]. Here a method for object identification with temporal dimension is presented. In our setting we could consider each cluster found in a snapshot as one observation of an object. We can then cluster observations from different snapshots in order to determine which senses are likely to belong to the same object and be evolutions of one another. An observation outside of a cluster can be considered similar to the sense represented by the cluster, but not as an evolved version of that sense.

A method for describing and tracking evolutions can be found in the related field of Temporal Text Mining. In [9] themes are found and tracked over time. A theme evolution graph is defined that seems particularly suitable for describing terminology evolution and is similar to what is proposed in [17]. To our knowledge only one previous work has been published in the area of Terminology evolution [1]. Here an orthogonal approach to ours is presented, where the aim is to find good query reformulations to concurrent language using language from the past. A term from a query can be reformulated with a similar term if the terms in the resulting query are also coherent and popular. Terms are considered similar if they co-occur with similar terms from their respective collections.

3   Problem Statement

In order to describe terminology evolution, we need a model. In [17] I developed an overall model which describes terminology evolution independently of what methods are used for solving each sub task.

3.1   Terminology Evolution Model

To represent the relation between terms and their meanings we introduce the notion of concept and represent meanings as connections between term and concept nodes in a graph. A concept represent a sense and can be seen as a synset used in WordNet [10]. To be able to describe terms and their meanings we define a *term concept graph* (TCG). In a TCG each term is associated with its concepts (word senses) found from a collection. The graph carries a sense of time by annotating each edge with the time stamp inherited from the collection.

$$\phi : W \times T \to (W \times \mathcal{P}(\mathcal{C} \times \mathcal{P}(T))) \tag{1}$$

$$(w,t) \mapsto (w, \{(c_1, \{t\}), \ldots, (c_n, \{t\})\})$$

where $w \in W$, $t \in T$ and for all $i = 1 \ldots n$: $c_i \in C$. Here $\mathcal{P}$ denotes a power set, i.e. the set of all subsets, W the universe of terms, C the universe of all clusters and T are time stamps. Although ϕ generates only one timestamp for each term-concept relation, we introduce the power set already here to simplify the next steps. We call the set of all TCGs derived from a collection for a *terminology snapshot*. Once several terminology snapshots have been constructed, these will need to be compared in order to find terminology evolution. This comparison is made by merging two TCGs and outputting a new TCG. The newly constructed TCG carries the gathered information of the merged TCGs by allowing for several time annotations on each edge. To shorten the notation we define τ as a set of time stamps t_i, i.e. $\tau \in \mathcal{P}(T)$ and a term concept relation can be written as a pair (c_i, τ_i). The main difficulty in the merging step comes from decisions concerning two similar clusters, see Fig. 1. When two clusters are different as in the case with c_2 and c_4 the merging is trivial. In the merged TCG the term w is associated with both c_2 and c_4 and the time annotations stay the same as in
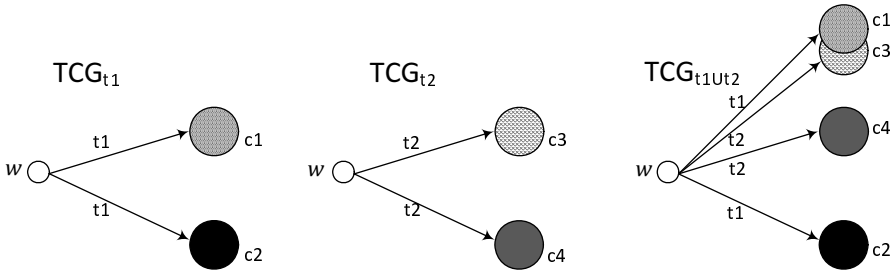
Fig. 1. Merging TCGs from t_1 and t_2 into one TCG containing all information about w until time t_2

the original TCGs. With c_1 and c_3, a decision must be made whether the differences come from the underlying collection or if c_1 evolved into c_3. The function representing the merging step can more formally be described by the following:

$$\psi : (W \times \mathcal{P}(\mathcal{C} \times \tau)) \times (W \times \mathcal{P}(\mathcal{C} \times \tau)) \rightarrow (W \times \mathcal{P}(\mathcal{C} \times \tau)) \qquad (2)$$

$$((w, \{(c_1, \{t\}), \ldots, (c_i, \{t\})\}), (w, \{(c_j, \tau_j), \ldots, (c_m, \tau_m)\}))$$
$$\mapsto (w, \{(c_1', \tau_1'), \ldots, (c_n', \tau_n')\})$$

where $c_i, c_j' \in \mathcal{C}, t \in T$ and $\tau_i, \tau_j' \in \tau$ for all i, j. It should be clear that the set of concepts c_i' in the resulting graph of ψ does not necessarily have to be a subset of the concepts $\{c_1, \ldots c_m\}$ from the input graphs, e.g. in Fig. 1, the concepts c_1 and c_3 could be merged (by union, etc.) and considered as a new concept. ψ can be iteratively applied to a TCG from time t_N and the TCG containing all knowledge about a term up to time t_{N-1}.

A TCG cannot directly be used for determining evolutions between terms and therefore we need a method for associating a concept with the relevant terms. For a given concept c, the function $\theta : C \rightarrow \mathcal{P}(W \times \tau)$ returns the set of terms used to express c, together with time stamp sets which denote when the respective terms were in use.

3.2 Research Hypothesis

In our terminology evolution model, we assume that there are some 'true' functions ϕ, ψ and θ that correctly and exhaustively models language from the underlying collections. Because these functions are unknown, automatically detecting terminology evolution becomes finding approximations for ϕ, ψ and θ.

4   Proposed Approach

The techniques which will be developed in this thesis for detecting terminology evolution will be divided into two main parts. The first will mainly concentrate

on finding word senses from a collection in an unsupervised manner, i.e. perform WSD including preprocessing data, terminology extraction and creation of terminology snapshots. The second task is to detect evolution in an automatic fashion and can be described as merging terminology snapshots to first find evolution of clusters and from this derive evolution of terms. For this thesis we will mainly focus on the second task and apply existing technologies for the first task to create input for task two.

4.1   Word Sense Discrimination

In this thesis we have chosen to include Terminology extraction as a subtask of WSD because it very much affects the results of the sense discrimination. We consider nouns as terms and once the relevant terms have been identified in a collection, a relational matrix is constructed. We have chosen the method presented in [3] which is a graph theoretical approach. Each node is a noun and there exists an edge between two nodes if they are separated by 'and', 'or' or commas in the collection. Following [3] we cluster the matrix using the curvature clustering algorithm with different clustering coefficients. These clusters are used to create TCGs. One issue to be tackled is to determine which terms from a concept that are associated with the concept, i.e. finding θ. Some terms are supporting terms for the concept while other terms are more important. In the case of fireman, the term "hose" might not be directly related to the concept and hence this member of the concept should not be pointing to the concept. Applying the clustering algorithm iteratively and as the clustering becomes stricter, monitoring which cluster members stay the same, can give us clues on which terms are more or less relevant for the cluster.

4.2   Tracking Evolution

Using the graph model to find concepts in the previous step has the advantage of clustering nodes with links. This means we can consider a concept to be more than a set of terms because we know how they relate to each other. Since two similar clusters can represent different senses if the nodes are linked differently, the links can aid the comparison. For tracking clusters we plan on using the approach presented in [12], which considers also link information between nodes. One interesting aspect is to see if clusters or communities representing word senses behave similarly to communities of co-authorships or telecommunication graphs. In [12] it is found that small communities should be stable in order to survive while larger communities need to be dynamic.

The main difference between tracking cluster evolution and tracking terminology evolution is the notion of members or nodes. Members are traditionally considered fixed; an author is always the same author even when its membership in clusters changes over time. When tracking concepts, it is not only word senses that change over time, but also the members themselves. E.g. the term "pretty" meant "cunning" and "crafty" and now means "attractive" and "good looking". To tackle this problem existing models need to be extended and adapted to our

purposes. Another large issue arises once we have found the evolution of concepts, i.e. we are able to track clusters over time. As discussed above, not all terms in a cluster are relevant to the cluster and hence we need to determine which terms from the concepts that can be considered evolutions of the target term in a TCG. One possible solution could be to introduce probabilities associated with cluster evolutions as well as between terms themselves. Then the probability for the target term in a TCG to evolve to a candidate term can be modeled using the cluster evolution probability as well as the internal 'importance' of a node to a cluster using a network flow graph.

5   Evaluation Methods

To evaluate the first step of the model, we evaluate the output of the word sense discrimination. We evaluate the clusters outputted by the curvature clustering against WordNet using techniques presented in [14] with tools from Word-Net::Similarity[2] package. To our knowledge no evaluation methods for the second step, i.e. automatic terminology evolution detection, have previously been published. Hence there are no reference datasets or measures to use. We plan on doing an evaluation on The Times corpus by manually choosing some examples of terms where there is evidence of evolution as well as terms without evolution in the archive. For each term we will verify if our terminology evolution algorithm can detect evolution in a way corresponding to the manual assessment. Success will be measured in the number of correctly assessed cases.

6   Results

Preliminary experiments are run on roughly 10% of The Times Archive, using 4 years of data every 50 years. For each collection we extract nouns, lemmatize them and build a graph as described in Sec. 4.1. An edge is kept if the corresponding nodes co-occure more than twice in the collection. We use a clustering coefficient of 0.5 which gives relatively stable senses [3,4,13]. We also use coefficient 0.3 hoping to get a broader coverage but also senses that are less stable and hence have higher probability of evolution. We can already see some interesting trends. Firstly the number of clusters corresponds well to the number of nodes (distinct nouns) found in the co-occurrence graph as seen in Fig. 2(a).

Secondly the graphs created for these years, all have high precision when evaluated using the approach from [14] with similarity threshold 0.25 [3,14]. In Fig. 2(b) we see a distinction between years $1785 - 1836$ and $1885 - 1985$. The higher precision in the early period is shown to be statistically significant with 95% confidence. The dip in this curve can partially be explained by the sizes of the collection. The same filtering is used regardless of the size of the collection. Instead filtering should be adjusted based on the collection size. e.g. for 1935, if

---

[2] A description of this package can be found on http://www.d.umn.edu/ tpederse/ similarity.html
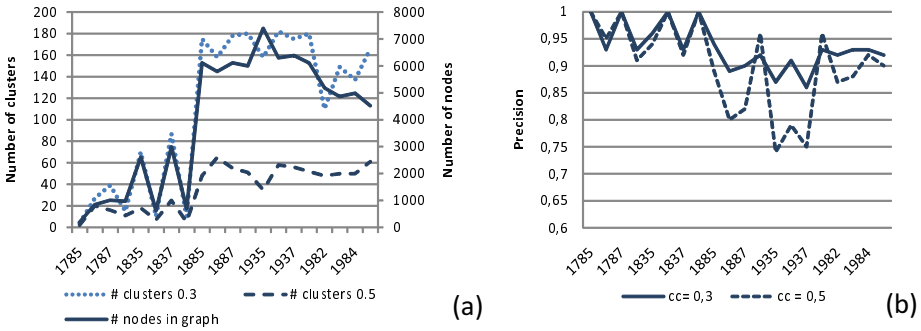
Fig. 2. Initial experiments using 10% of The Times Archive, where cc is the clustering coefficient used

an edge is kept with a frequency above 6, the precision of the clusters increases to 0.95 and 0.96 for $cc = 0.3$ and 0.5 respectively. The distinction between early and later years is shown for the number of distinct nodes in the graphs and the number of clusters using both coefficients. It is also the case that during this period we find a higher amount of WordNet terms among all extracted terms than in the latter period, also this with a 95% confidence level. This is interesting because the behavior cannot be seen in the proportion of nouns extracted from all terms from the collections.

7   Discussion

One of the larger difficulties in detecting terminology evolution is determining if two similar clusters should be considered the same or evolutions from each other. The main difficulty lies in that we are only making approximations of word senses and these rely very much on the underlying collection. Assume we have 100 documents describing the term "rock", some describing the "music" sense of rock while others describing the "stone" sense. Applying WSD techniques on several random partitioning into batches of 50 documents would yield in different results for each partitioning. If a human expert assessed two clusters on the same sense from two different partitioning, it would be a fairly easy decision, even with some missing terms in one cluster. While for automatic detection of terminology evolution, this is a more difficult decision to make. When are the differences between two clusters dependent on the underlying collection, and when has there been evolution?

Another issue arises when using the TCG to find and describe evolution. Describing terminology evolution like in the case with "rock" is straight forward given the TCG model. A concept representing the new sense of the term is added and the appropriate timestamp is given to the edge between the concept and the term. The "St. Petersburg" example on the other hand is more complicated. It must first be determined that the concept nodes associated to the terms "Leningrad", "Petrograd" and "St. Petersburg" are in fact the same and from this arrive at the fact that "Leningrad" and "Petrograd" can be used to

expand the query "St. Petersburg". The methodology for this could invite many 'false positive'."Sean Penn" and "Guy Ritchie" are both terms that would be associated with the concept node representing "Madonna", both being husbands of her in different periods of time. If the model can identify the "Madonna" node as representing the same concept for both these terms, then we could draw the conclusion that "Sean Penn" can be expanded with "Guy Ritchie" in the same way as with the "St. Petersburg" example.

8 Conclusions and Future Work

As discussed above, my Ph.D. thesis aims at finding ways to automatically detect terminology evolution. We have identified the first step in this process to be automatic detection of word senses from a collection. The second step is to discover evolution of these senses and from this dervie at evolution of terms. So far the initial experiments have shown that the performance of the WSD algorithm chosen, is stable over time. The word senses found by the algorithm map well to synsets from WordNet and with a low clustering coefficient, we are able to get a higher coverage of the collection. We will conduct large scale experiments on The Times collection to get a more comprehensive overwiev. The next step for this thesis will be to determine which terms are relevant to the concepts in order to create TCGs. When this first task has been solved we will continue with tracking evolution of clusters representing word senses. Here we will take into account the linkage information between the nodes in the clusters in order to better map senses from different periods in time. We plan on adapting current algorithms for tracking clusters to better fit the terminology evolution scenario as discussed above. We will conduct experiments to monitor the life span of clusters to see if the same properties hold for word senses as for network clusters. The last step of the process will be to dervie evolution of terms from the word sense evolutions found in the previous step. The technologies used are very much dependent on the outcome of, as well as technologies used in, the previous steps. One possibility is to consider cluster evolution probabilities, as well as term importance in a cluster, to calculate flows using a graph flow model and from these flows derive at terminology evolution.

Acknowledgements

References

1. Berberich, K., Bedathur, S., Sozio, M., Wiekum, G.: Bridging the terminology gap in web archive search. In: WebDB (2009)
2. Deschacht, K., Francine Moens, M., Law, I.C.F.: Text analysis for automatic image annotation. In: Proc. of the 45 th Annual Meeting of the Association for Computational Linguistics. East Stroudsburg (2007)

3. Dorow, B.: A graph model for words and their meanings. PhD thesis, University of Stuttgart (2007)
4. Dorow, B., Widdows, D., Ling, K., Eckmann, J.P., Serqi, D., Moses, E.: Using curvature and Markov clustering in graphs for lexical acquisi tion and word sense discrimination. In: 2nd Workshop organized by the MEANING Project, Trento, Italy, February 3-4 (2005)
5. Ferret, O.: Discovering word senses from a network of lexical cooccurrences. In: Proc. of the 20th international conference on Computational Linguistics, Morristown, NJ, USA, ACL, p. 1326 (2004)
6. Lin, D.: Using syntactic dependency as local context to resolve word sense ambiguity. In: Proc. of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, Morristown, NJ, USA, ACL, pp. 64–71 (1997)
7. Lin, D.: Automatic retrieval and clustering of similar words. In: Proc. of the 17th international conference on Computational linguistics, Morristown, NJ, USA, ACL, pp. 768–774 (1998)
8. Lin, Y.R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In: Proc. of the 17th international conference on World Wide Web, pp. 685–694. ACM, New York (2008)
9. Mei, Q., Zhai, C.: Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In: Proc. of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 198–207. ACM Press, New York (2005)
10. Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM 38, 39–41 (1995)
11. Oyama, S., Shirasuna, K., Tanaka, K.: Identification of time-varying objects on the web. In: Proc. of the 8th ACM/IEEE-CS joint conference on Digital libraries, pp. 285–294. ACM, New York (2008)
12. Palla, G., Barabasi, A.L., Vicsek, T.: Quantifying social group evolution. Nature 446(7136), 664–667 (2007)
13. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 814–818
14. Pantel, P., Lin, D.: Discovering word senses from text. In: Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 613–619 (2002)
15. Schütze, H.: Automatic word sense discrimination. Journal of Computational Linguistics 24, 97–123 (1998)
16. Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., Schult, R.: Monic: modeling and monitoring cluster transitions. In: Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 706–711. ACM, New York (2006)
17. Tahmasebi, N., Iofciu, T., Risse, T., Niederee, C., Siberski, W.: Terminology evolution in web archiving: Open issues. In: Proc. of 8th International Web Archiving Workshop in conjunction with ECDL (2008)