Natural Language Engineering **21** (5): 773–798. © Cambridge University Press 2015
doi:10.1017/S135132491500011X

An automatic approach to identify word sense changes in text media across timescales

SUNNY MITRA[1], RITWIK MITRA[1], SUMAN KALYAN MAITY[1], MARTIN RIEDL[2], CHRIS BIEMANN[2], PAWAN GOYAL[1] and ANIMESH MUKHERJEE[1]

[1]*Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India*
e-mail: {sunnym,ritwikm,sumankalyan.maity,pawang,animeshm}@cse.iitkgp.ernet.in
[2]*FG Language Technology, Computer Science Department, TU Darmstadt, Darmstadt, Germany*
e-mail: {riedl,biem}@cs.tu-darmstadt.de

(*Received 30 May 2014; revised 29 January 2015; accepted 30 January 2015; first published online 16 April 2015*)

Abstract

In this paper, we propose an unsupervised and automated method to identify noun sense changes based on rigorous analysis of time-varying text data available in the form of millions of digitized books and millions of tweets posted per day. We construct distributional-thesauri-based networks from data at different time points and cluster each of them separately to obtain word-centric sense clusters corresponding to the different time points. Subsequently, we propose a split/join based approach to compare the sense clusters at two different time points to find if there is 'birth' of a new sense. The approach also helps us to find if an older sense was 'split' into more than one sense or a newer sense has been formed from the 'join' of older senses or a particular sense has undergone 'death'. We use this completely unsupervised approach (a) within the Google books data to identify word sense differences within a media, and (b) across Google books and Twitter data to identify differences in word sense distribution across different media. We conduct a thorough evaluation of the proposed methodology both manually as well as through comparison with WordNet.

1 Introduction

Word meanings are not fixed; instead, they undergo changes either due to the advent of new word senses or due to established word senses taking new shades of meaning or becoming obsolete. In principle, word senses may expand/become more generalized including more referents; may contract or narrow down to include fewer referents; may shift/transfer to include a new set of referents. For example, the word 'barn' referred to 'barley storage' earlier while it now refers to 'large shed for railroad cars/truck *etc.*' *i.e.* the sense of the word has broadened, on the other hand, the word 'liquor' earlier meant 'fluid' and is now narrowed to only 'alcohol'. Another interesting aspect of word sense change arises due to the presence of polysemous words. These words take various meanings while appearing in different contexts. For instance, the word 'bank' has several distinct interpretations, including that of a ‘financial institution’ and the ‘shore of a river’. Automatic discovery and disambiguation of word senses from a given text is an important and challenging problem that has been extensively studied in the literature (Spärk-Jones 1986; Ide and Veronis 1998; Schütze 1998; Navigli 2009). However, an equally important aspect that has not been so far well investigated corresponds to one or more changes in the range of meanings expressed by a word. This particular aspect is getting increasingly attainable as more and more diachronic text data are becoming available in the form of millions of tweets posted per day[1] on online social networks like Twitter or through millions of digitized books (Goldberg and Orwant 2013) published over the last centuries. As a motivating example one could consider the word ‘sick’ – while according to the standard English dictionaries this word usually refers to some illness, a new meaning of ‘sick’ referring to something that is ‘crazy’ or ‘cool’ is currently becoming popular in the English vernacular. This change is further interesting because while traditionally ‘sick’ has a negative sense, the current meaning stands positive.

Ever since the emergence of human communication, words have gone through sense changes (Bamman and Crane 2011; Michel *et al.* 2011; Wijaya and Yeniterzi 2011; Mihalcea and Nastase 2012); however, with the advent of modern technology and the availability of huge volumes of diachronic data, this research avenue has broadened and so have its applications. Many Natural Language Processing (NLP) tasks like Q&A or Machine Translation depend on lexicons for the part-of-speech (POS) or meaning representation of a word. If a sense of a word is not found in a system’s lexicon, the system typically fails to recognize the novel word sense and performs erroneous inference and the overall performance of the entire system is therefore likely to suffer due to this incorrect lexical information. Therefore, automatically identifying novel word senses has become an important and challenging task in lexical acquisition. Improved methodologies on automatic tracking of sense changes can help the lexicographers in word sense discovery, and researchers in enhancing various NLP/Information Retrieval (IR) applications (*e.g.* disambiguation, semantic search, *etc.*) that are naturally sensitive to change in word senses.

The above motivation forms the basis of the central objective set in this paper, which is to devise a completely unsupervised approach to track noun sense changes in large texts available over multiple timescales and over two media. Toward this objective we make the following contributions: (a) extend a graph clustering-based sense induction algorithm (Biemann 2006) on diachronic data, (b) use the diachronic sense clusters to develop a split-join based approach for identifying new senses of a word, and (c) evaluate the performance of the algorithms on various datasets using different suitable manual and automated methods. Comparison with the English WordNet indicates that in 51% of the cases from a representative sample within the Google books data (1909–1953 *versus* 2002–2005), there has been a birth of a completely novel sense. While our main concern was to detect ‘birth’ of a new sense, the proposed approach is general enough to detect ‘split’ and ‘join’ of senses

[1] Roughly 500 million tweets per day, source `http://www.internetlivestats.com/twitter-statistics/`

as well. Over this sample, an evaluation based on WordNet indicates that in 46% cases a new sense has split off from an older sense and in 63% cases two or more older senses have merged in to form a new sense. In case of Books *versus* Twitter comparison, the average of birth cases verifiable via WordNet is roughly 42–47% across various samples.

The work presented here is an extension of Mitra *et al.* (2014). The novel aspects and contributions of this paper with respect to the conference version are (a) it is an extended version of the conference paper with a detailed explanation of the proposed methodology with illustrative examples and (b) in addition to the Google books dataset, we also use a corpus from Twitter in the experiments, adding a comparison of senses across different media.

The remainder of the paper is organized as follows. In the next section, we present a short review of relevant literature. In Section 3, we describe the datasets used for this study and outline the process of distributional-thesaurus-based network construction in detail. In Section 4, we present an approach based on graph clustering to identify the diachronic sense clusters and in Section 5, we present the split-join based framework to track word sense changes. Experimental methods are detailed in Section 6. The evaluation framework for both the manual and automated evaluation are described and results are presented in Section 7. Finally, conclusions and further research directions are outlined in Section 8.

2 Related work

Word sense disambiguation and word sense identification have both remained key areas right from the very early initiatives in the natural language processing research. Ide and Veronis (1998) present a very concise survey of the history of ideas used in word sense disambiguation; for a recent survey of the state of the art one can refer to Navigli (2009). Some of the first attempts to automatic word sense discovery were made by Spärk-Jones (1986); later in lexicography, it has been extensively used as a pre-processing step for preparing mono- and multi-lingual dictionaries (Kilgarriff and Tugwell 2001; Kilgarriff *et al.* 2004). However, none of these works consider the temporal aspect of the problem.

In contrast, the current study is inspired by the works on language dynamics and opinion spreading (Mukherjee *et al.* 2011; Maity, Venkat and Mukherjee 2012; Loreto, Mukherjee and Tria 2012) and automatic topic detection and tracking (Allan, Papka and Lavrenko 1998). However, our work differs significantly from those proposed in the above studies. Opinion formation deals with the self-organization and emergence of shared vocabularies, whereas our work focuses on how the different senses of these vocabulary words change over time and thus become ‘out of vocabulary’. Topic detection involves detecting the occurrence of a new event such as a plane crash, a murder, a jury trial result, or a political scandal in a stream of news stories from multiple sources, while tracking is the process of monitoring a stream of news stories to find those that track (or discuss) the same event. This is done on shorter timescales (hours, days), whereas our study focuses on larger timescales (decades, centuries) and we are interested in common nouns as opposed to events, which are characterized mostly by the named entities. Blei and Lafferty (2006) used a dataset spanning 100 years from *Science* and using dynamic topic modeling, to analyze the time evolution of topics. Wang and McCallum (2006) used 17 years of *NIPS* research papers and 200 years of presidential addresses for modeling topics over time. In dynamic topic modeling, the distribution of words associated with a topic change over time. In contrast, our method attempts to identify changes in the sense of each target word as opposed to a topic, which is a probability distribution over the vocabulary. Google books n-gram viewer[2] is a phrase-usage graphing tool which charts the yearly count of selected letter combinations, words or phrases as found in over 3.4 million digitized books. It only reports frequency of word usage over the years, but does not give any correlation among them as *e.g.* in Heyer, Holz and Teresniak (2009), and does not analyze their senses.

A few approaches suggested in Bond *et al.* 2009 and Pääkkö and Lindén (2012) attempt to augment WordNet synsets primarily using methods of manual annotation. Cook and Stevenson (2010) use corpora from different time periods to study the change in the semantic orientation of words. Gulordava and Baroni (2011) used two different time periods in the Google n-grams corpus and presented an approach to detect semantic change based on distributional similarity between word vectors. Another recent work by Cook *et al.* (2013) attempts to induce word senses and then identify novel senses by comparing two different corpora: the 'focus corpora' (*i.e.* a recent version of the corpora) and the 'reference corpora' (older version of the corpora). However, these methods are either based on supervised annotation schemes or are conducted over only two time points. This stands in contrast to our approach, which utilizes several (here: eight) time-points, thus allowing us to perform a detailed stability analysis of the sense changes, reported for the first time in this paper. One of the closest works to what we present here has been put forward by Tahmasebi, Risse and Dietze (2011), where the authors track senses in a newspaper corpus containing articles between 1785 and 1985.

With our work, we address the following limitations of previous work: First, our method does not compare only two corpora, but several corpora from different time spans, which allows us to more closely track the point in time when a sense change has occurred and also yields more stable results. Further, we address not only new senses, but also cases where two senses become indistinguishable ('join'), one sense splits into several senses, or a sense falls out of the vocabulary ('death'). Further, we provide a thorough evaluation procedure and assess our results not only manually, but also automatically with the help of WordNet. We introduce, for the first time, a completely unsupervised and automatic method to identify the change of a word sense across multiple media and over large timescales. In addition, our scheme allows us to correctly identify the stable sense changes.

3 Datasets and graph construction

It is well known that context plays a crucial role while identifying the sense of a word. According to the *distributional hypothesis*, 'a word is characterized by the

[2] https://books.google.com/ngrams

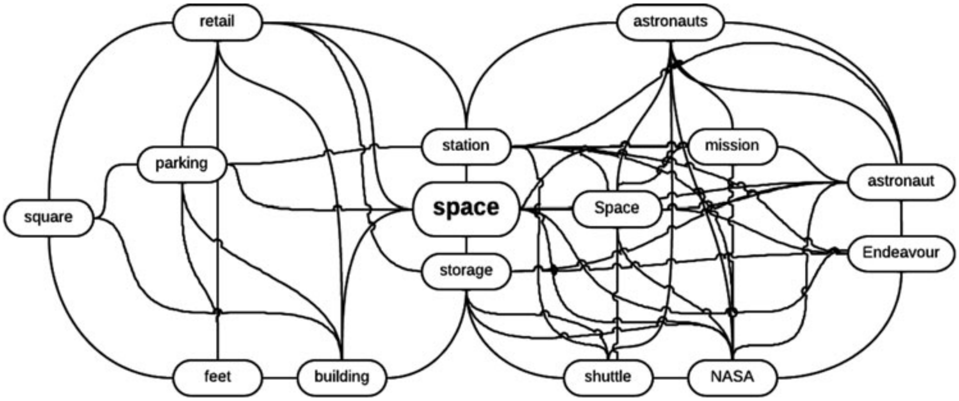

Fig. 1. Word co-occurrence network for the word 'space'.

company it keeps' (Firth 1957). Figure 1 shows a word co-occurrence graph[3] for the word 'space' (since graph is corpus-dependent). There are two sets of neighboring words around the word 'space': the left one signifying 'office space' and the right one signifying 'outer space'.

We exclusively use such co-occurrence based networks across different timescales to track sense change of a word. For preparing such a network, we have used two different datasets: (a) Google books syntactic n-grams, and (b) Random tweets from Twitter.

3.1 Google books syntactic n-grams

This dataset is based on Google English Books corpus. The corpus consists of texts from over 3.4 million digitized English books. While the dataset contains books published between 1520 and 2008, most of them were published after 1800. The corpus is also available in several subsets: Uniformly sampled 1 M English books, Works of Fiction, American English books published in the US, British English books published in Britain, *etc*.

For a detailed understanding on how this dataset is prepared from the above corpus, the reader is referred to Goldberg and Orwant (2013). The format of the dataset is as follows. Each line represents one syntactic n-gram. A line is of the form: **head_word** [TAB] **syntactic n-gram** [TAB] **total_count** [TAB] **counts_by_year**, where the **counts_by_year** is a tab-separated list of **year** [COMMA] **count** items, and the **syntactic n-gram** is a space-separated list of tokens and each token format has the form 'word/postag/deplabel/headindex'. We utilized the arcs in Google syntactic n-grams, which represent direct dependencies between two content words and reflects in most cases a syntactic bigram, cf. Riedl, Steuer and Biemann (2014).

Example of a syntactic bigram:

```
data data/NNS/pobj/0 acquisition/NN/conj/1
```

[3] In a word co-occurrence graph, words are denoted by nodes, and there exists an edge between two nodes, if the corresponding words co-occur in a sentence.

Example of a complete line:

```
data data/NNS/pobj/0 acquisition/NN/conj/1 15 1974,1 1980,2 1985,1
        1988,2 1989,1 1990,1 1991,2 2002,2 2006,2 2007,1
```

3.2 Random tweets from Twitter

This dataset is based on millions of tweets posted over Twitter. The corpus consists of a random sample of 1% of the Twitter data for the years 2012 and 2013, collected via the Twitter streaming API[4] for the years 2012 and 2013, which was filtered further to use tweets in English only. We generated positional bigrams, *i.e.* two words are connected by an arc if they are observed next to each other. Tweets were not normalized, since we did not want to conflate results of our algorithm with artifacts caused by the normalization. Besides, since the processing of tweets is based on n-grams and does not rely on linguistically informed pre-processing steps, normalization was not deemed necessary. Additionally, no part-of-speech tagging and lemmatization was used for the Twitter data.

3.3 Graph construction

Initially both our datasets are in the form of (syntactic or positional) bigrams. However, we use these bigrams in order to construct a distributional thesaurus (henceforward abbreviated DT) (Lin 1997; Rychlý and Kilgarriff 2007) that contains for each word a list of words that are similar with respect to their bigram distribution. As our datasets are divided across different time periods, we prepare a separate DT-based network for each of these time periods. We briefly outline the procedure of constructing the DT-based network in the following sections. For a detailed description, please refer to Biemann and Riedl (2013).

3.4 Distributional thesaurus-based network

For DT construction, we proceed along the following steps. We compute the LMI[5] (Evert 2005) for each bigram, which gives a measure of the collocational strength of a bigram. Each (syntactic or positional) bigram is broken into a word and a feature, where the feature consists of the (syntactic or positional) bigram relation and the related word.

Then we retain top ranked 1,000 features for each word. Finally, for each word pair, we obtain the intersection of their corresponding feature set. If the overlap is above a threshold, we retain the pair in the DT-based network, setting the edge weight to the number of overlapping features. The LMI measure was shown to yield the best results amongst several measures for feature ranking in this approach in Biemann and Riedl (2013).

[4] https://dev.twitter.com/streaming/public

[5] Lexicographer's Mutual Information (LMI):
$LMI(word, feature) = f(word, feature)\log_2(\frac{f(word,feature)}{f(word)f(feature)})$

4 Unsupervised sense induction

In this section, we present our completely unsupervised technique for identifying different senses of a word. According to Figure 1, there are two sets or clusters forming around the word 'space' signifying two different senses for the same. Likewise, if we can identify all the sense clusters from our DT-based networks, our requirement is fulfilled. Hence, we need a graph-clustering framework. We have used Chinese Whispers (CW) graph clustering as introduced in (Biemann 2006). For the purpose of readability we briefly outline the basic steps that are followed to obtain the sense clusters. For a more formal description and analysis, the reader is referred to Biemann (2012).

Neighborhood graph construction. As a first step, we consider each word in the DT-based network and call it a target word. Next, we construct a word graph around every target word based on the similar words found in the DT-based network; this is also termed as the ego or the open neighborhood of the target word (Biemann 2012). The open neighborhood is defined in terms of two parameters: N and n – only the most similar N words of the target enter the graph as nodes, and an edge between nodes is drawn only if one of the corresponding words is contained in the most similar n words of the other. Further, for the entire analysis we remove those edges from the DT-based network that have very low edge weights (assumed to be ≤ 5 for this study).

Clustering the neighborhood graph. The neighborhood graph is clustered using the CW algorithm (Biemann 2006). The algorithm works in a bottom-up fashion as follows: initially, all nodes are assigned to different clusters. Then the nodes are processed in a random order for a small number of iterations and inherit the predominant cluster in the local neighborhood. This is the cluster with the maximum sum of edge weights to the current node under consideration, where edge weights are optionally downweighted by the degree of the neighbor. In case of multiple predominant clusters, one is chosen randomly. In general, the algorithm has been empirically shown to converge within a few iterations producing the desired clusters. During clustering, the individual nodes can be further assigned weights in three different ways – (a) dividing the influence of a vertex in the update step by the degree of the vertex, (b) dividing by the natural logarithm of the degree + 1 and (c) not doing vertex weighting – exactly as described in Biemann (2012).

Collecting the clusters. The algorithm produces a set of clusters for each target word by organizing its open neighborhood into clusters. We hypothesize that each different cluster corresponds to a particular sense of the target word. We use these clusters, and in particular, observe how they change over time for a given target word. We further apply the same observation for algorithmic identification of sense changes in the next section.

Some important properties of CW that are worth mentioning here are:

Non-determinism: CW is non-deterministic in nature. If we run the CW algorithm multiple times, it may produce different clusters affecting our final outputs. To overcome this issue, we have included a few filtering techniques, described in Section 5.6 of this paper.

Overlapping clusters: CW produces overlapping clusters, *e.g.* the word ‘beautiful’ may be present in the CW clusters of ‘girl’ and ‘painting’ simultaneously.

While the evaluation of unsupervised sense induction systems is inherently difficult, the utility of the system discussed here has been demonstrated to significantly increase the performance of word sense disambiguation (Biemann 2010) and lexical substitution (Biemann 2012) when used as a feature in a supervised machine learning setting. While our methodology fails to detect extremely rare senses due to the application of various thresholds described above, we have observed that it is capable of finding up to a dozen senses for highly ambiguous words, many of them are rare.

5 Tracking sense changes

This section presents an algorithmic procedure to track sense change of a word by comparing the sense clusters of two different time periods. Let us consider that we are comparing the sense clusters of a word w between two different time intervals, tv_i and tv_j, where tv_i is the older time period between the two. Let us assume, for the word w, we have found m sense clusters, namely $\{s_{i1}, s_{i2}, \ldots, s_{im}\}$, in tv_i and n sense clusters, namely $\{s_{j1}, s_{j2}, \ldots, s_{jn}\})$ in tv_j from the CW algorithm, where s_{xy} denotes yth sense cluster during time interval tv_x. Next, we describe the procedure for detecting a sense change by comparing these clusters.

5.1 Split, join, birth, and death

If there is a change in the cluster set of a word from one time period to another, the word may have undergone a sense change during the time interval in between. During this change, the structure of few older clusters may change through splitting or merging, or a totally new cluster containing words that were not neighbors before may appear suddenly, or even an older cluster may vanish gradually. Therefore, we propose that a word w can undergo sense change from one time period (tv_i) to another (tv_j) if any of the following occurs:

Split: A sense cluster (s_{ix}) of older time period (tv_i) evenly splits into two clusters (s_{jy} and s_{jz}) in the newer time period (tv_j). Formally $s_{ix} = s_{jy} \cup s_{jz}$.

Join: Two sense clusters (s_{ix} and s_{iy}) of older time period (tv_i) get merged into a single cluster (s_{jz}) in newer time period (tv_j). Formally $s_{jz} = s_{ix} \cup s_{iy}$.

Birth: A new sense cluster (s_{jy}) appears in newer time period (tv_j) but was not present in the older time period (tv_i). Thus, s_{jy} contains words that were not neighbors of w in tv_i, *i.e.* $\forall k \in [1, m], s_{ik} \cap s_{jy} = \emptyset$

Death: A sense cluster (s_{ix}) in older time period (tv_i) vanishes and does not appear in the newer time period (tv_j). Formally $\forall k \in [1, n], s_{ix} \cap s_{jk} = \emptyset$

In Figure 2, we show a schematic diagram illustrating *split*, *join*, *birth* and *death.*

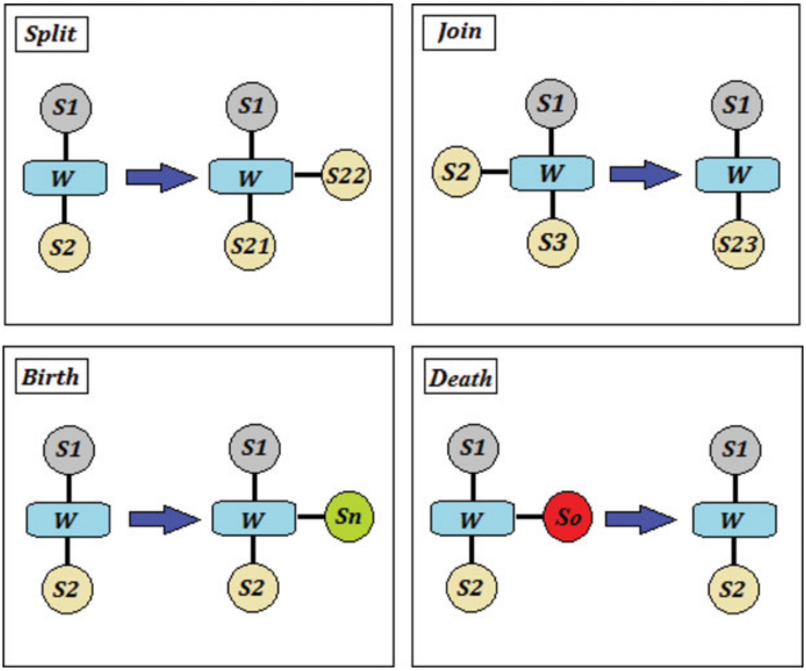

Fig. 2. (Colour online) Schematic diagram illustrating *split*, *join*, *birth,* and *death.*

5.2 *Data structure*

In our algorithm to detect split, join, birth, or death, we create a two-dimensional matrix, I, of size $(m+1) \times (n+1)$. We call it the 'intersection table'. The first m rows correspond to the m sense clusters of the word w in tv_i and first n columns correspond to the n sense clusters of w in tv_j. An element in this range signifies the number of words present in both the corresponding sense clusters (*i.e.* intersection). We keep an extra row to capture the number of words in the corresponding sense clusters in tv_j that did not appear in any of the sense clusters of tv_i. Similarly, we keep an extra column to capture the number of words in the corresponding sense clusters in tv_i that did not appear in any of the sense clusters of tv_j. Hence, an element in the intersection table is defined as follows:

$$I_{xy} = \begin{cases} |s_{ix} \cap s_{jy}|, & \text{if } (1 \leq x \leq m) \text{ and } (1 \leq y \leq n). \\ |s_{jy} - \bigcup_k s_{ik}|, & \text{if } x = m+1 \text{ and } (1 \leq y \leq n). \\ |s_{ix} - \bigcup_k s_{jk}|, & \text{if } (1 \leq x \leq m) \text{ and } y = n+1. \end{cases}$$

To capture all the four possible scenarios for sense change, we convert the elements of intersection table into fractions with respect to the corresponding cluster sizes of either tv_i or tv_j depending on our need. Specifically, to detect birth or join we compute the fractions with respect to the cluster sizes of the newer time period, and to detect death or split we compute them with respect to the cluster sizes of the older time period.

Table 1. *Number of candidate birth senses within the Google books data for* 'compiler'

Time-period	Cluster ID	Words
1909–1953	C_{11}	*publishing, collection, editions, text, compilers, reprint, revision, author, copies, edition, authenticity …*
	C_{12}	*novelist, poet, illustrator, proprietor, moralist, auditor, correspondent, reporter, editor, dramatist …*
2002–2005	C_{21}	*administrator, clinician, listener, viewer, observer, statesman, teacher, analyst, planner, technician …*
	C_{22}	*implementations, controller, program, preprocessor, api, application, specification, architecture …*

5.3 Algorithm

After preparing the intersection table, we identify the four difference cases as follows:

- $\exists k \in [1,m]$, $\exists l \in [1,n]$, $\exists l' \in [1,n]$, $\frac{I_{kl}}{|s_{ik}|} \geq A_1$, $\frac{I_{kl'}}{|s_{ik}|} \geq A_1 \Rightarrow$ **split**. In other words, if there exists a row in the intersection table with two fractions $\geq A_1$ each, then it is a split.
- $\exists k \in [1,m]$, $\exists k' \in [1,m]$, $\exists l \in [1,n]$, $\frac{I_{kl}}{|s_{jl}|} \geq A_1$, $\frac{I_{k'l}}{|s_{jl}|} \geq A_1 \Rightarrow$ **join**. This means, if there exists a column in the intersection table with two fractions $\geq A_1$ each, then it is a join.
- $\exists l \in [1,n]$, $k = m+1$, $\frac{I_{kl}}{|s_{jl}|} \geq A_2 \Rightarrow$ **birth**. This means that if there exists a fraction in the additional row with value $\geq A_2$, then it is a birth.
- $\exists k \in [1,m]$, $l = n+1$, $\frac{I_{kl}}{|s_{ik}|} \geq A_2 \Rightarrow$ **death**. In this case, if there exists a fraction in the additional column with value $\geq A_2$, then it is a death.

Since we cannot expect a perfect split/join/birth/death, we use A_1 and A_2 as two parameters to denote the threshold values in our algorithm.

5.4 Illustration

We illustrate the working of our algorithm by considering the sense clusters of word 'compiler' from time periods 1909–1953 (earlier) and 2002–2005 (later). Some of the words in these sense clusters are shown in Table 1. For the earlier period of 1909–1953, we have two clusters (C_{11}, C_{12}), whose sizes are 35 and 64, respectively. Similarly for the later period 2002–2005, we have two clusters (C_{21}, C_{22}) having sizes 15 and 77, respectively.

We can use these sense clusters to construct the intersection table as shown in Table 2. As discussed in Section 5.2, the dimension of the table should be $(2+1) \times (2+1)$, *i.e.* , 3×3. Let I be the table, and I_{mn} is one of its cells from mth row and nth column. Originally, I_{mn} contains the size of the intersection of C_{1m} with C_{2n} for $1 \leq m \leq 2$ and $1 \leq n \leq 2$. The extra cells of the third row contain the number of elements in the corresponding clusters of the later period that do not appear in any of the clusters of the earlier period; similarly, the extra cells of the

Table 2. *Intersection table corresponding to sense clusters for* 'compiler' *from Table 1: fractions are shown with respect to the clusters of the later period*

	C_{21} (size 15)	C_{22} (size 77)	
C_{11} (size 35)	0 (0%)	1 (3%)	34
C_{12} (size 64)	10 (66%)	2 (3%)	62
	5 (34%)	74 (94%)	...

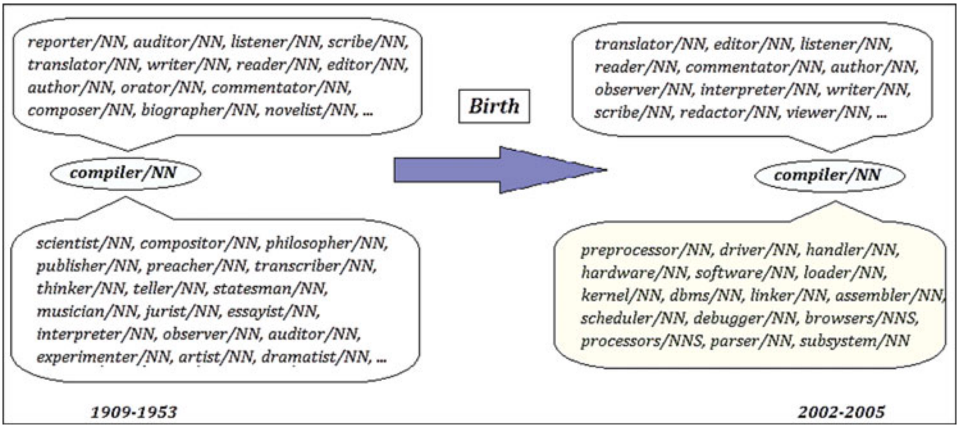
reporter/NN, auditor/NN, listener/NN, scribe/NN, translator/NN, writer/NN, reader/NN, editor/NN, author/NN, orator/NN, commentator/NN, composer/NN, biographer/NN, novelist/NN, ...
compiler/NN
scientist/NN, compositor/NN, philosopher/NN, publisher/NN, preacher/NN, transcriber/NN, thinker/NN, teller/NN, statesman/NN, musician/NN, jurist/NN, essayist/NN, interpreter/NN, observer/NN, auditor/NN, experimenter/NN, artist/NN, dramatist/NN, ...
1909-1953
Birth
translator/NN, editor/NN, listener/NN, reader/NN, commentator/NN, author/NN, observer/NN, interpreter/NN, writer/NN, scribe/NN, redactor/NN, viewer/NN, ...
compiler/NN
preprocessor/NN, driver/NN, handler/NN, hardware/NN, software/NN, loader/NN, kernel/NN, dbms/NN, linker/NN, assembler/NN, scheduler/NN, debugger/NN, browsers/NNS, processors/NNS, parser/NN, subsystem/NN
2002-2005

Fig. 3. (Colour online) Example of the birth of a new sense for the word 'compilers' by comparing 1909–1953 sense clusters with 2002–2005.

third column contain the number of elements in the corresponding clusters of the later period which are absent from all the clusters of the earlier period.

After finding all the counts, we need to convert the content of each cell to fractions. It is intuitive that for identifying *birth* or *join* case, these fractions have to be obtained with respect to the cluster sizes of the later period. Similarly, for identifying *split* or *death*, the fractions have to be obtained with respect to the cluster sizes of the earlier period. In Table 2, we show the fractions with respect to the later period. The percentage intersection of C_{22} with each of C_{11} and C_{12} is roughly 3% each and 94% of the words in this cluster are new. Therefore, we can consider the second cluster in 2002–2005 as the birth of a new sense. On the other hand, 66% of the words in C_{21} are contained in C_{12} and only 34% words are new. Therefore, C_{21} does not qualify as a birth cluster.

In Figure 3, we illustrate the birth of a new sense for 'compilers' using the graphical representation.

5.5 Time complexity

In our split/join based comparison algorithm, for each word in the later period we locate the same in the earlier period through a linear search. Then we compare all pairs of clusters of that word across these two time periods by taking intersection. Thus, the time complexity of the algorithm is $\Theta(w_1 w_2 mn(s_1 + s_2))$, where w_1 is the number of words in the earlier period, w_2 is the number of words in the later period, m is the average number of clusters of a word in the earlier period, n is the average number of clusters of a word in the later period, s_1 is the average cluster size in the earlier period, and s_2 is the average cluster size in the later period. The term $(s_1 + s_2)$ is appearing due to computation of the intersections of clusters.

5.6 Multi-stage filtering

The non-deterministic nature of the CW algorithm might produce different clusterings in different runs, which might affect subsequent processing. While we have not observed entirely random deviations due to this non-determinism, a common thing to note is that when repeating the clustering on the same graph, sometimes large clusters are broken into smaller ones that correspond to finer-grained aspects of meaning or usage (*e.g.* body part 'hip' as undergoing an examination *versus* as undergoing a surgery[6]). Since this is a critical issue when tracking splits and joins of clusters across time periods, we address this by running the clustering algorithm several times, see below. Apart from that, we include a few more filtering techniques to get the most meaningful portion out of our result. The following techniques are used in stages:

Stage 1. We execute the CW algorithm thrice on the DT-based network of the earlier as well as the later period. Thus we get three pairs of cluster sets from the three runs. Then we apply our split/join algorithm on each pair to obtain three candidate word lists. Finally we take those candidates from these three lists which appear in majority of them, *i.e.* we will take only those words that appear in at least two of the lists. Then we feed the final list obtained through this stage in the next one. We found that three runs were sufficient to rule out most of the instabilities caused by the non-determinism of CW.

Stage 2. As we focus on sense change of noun words for this experiment, we retain only those candidates that have a part-of-speech POS tag 'NN' or 'NNS'. Our Google books dataset was POS tagged, but the Twitter dataset was not. For the Twitter dataset, after getting all the candidate words we tag each of them according to the corresponding POS tag obtained for the Google books data and then retain only those words having 'NN' or 'NNS' tag (corresponding to a lexicon lookup).

Stage 3. After getting all the noun candidates from the previous stage, we sort them according to their frequency in the previous time period. Then, we take the *torso* (60%) of the frequency distribution from this list by removing the top 20% and the bottom 20% from it. Generally, these middle frequencies are the most discriminative words, and the most interesting for our analysis cf. (Luhn 1958; Kwong 1998). For the words in the low frequency range, there may not be sufficient evidence in the dataset to detect a sense change and rare words usually only have a single sense. On the other hand, words in

[6] Biemann (2012), pp. 146.

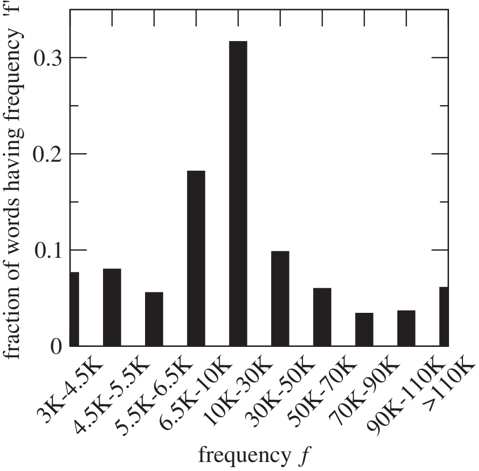

Fig. 4. Frequency histogram for candidate words while comparing 1909–1953 sense clusters with 2002–2005.

the high-frequency range tend to be less topic-oriented and thus, appear in very different contexts even when conveying the same (mostly abstract) sense, which resulted in too coarse-grained sense clusters in preliminary experiments since these high-frequency terms bridged otherwise unrelated clusters.

A frequency histogram for the candidate words obtained after Stage 2 is shown in Figure 4. The bottom 20% words belong to frequency range $3K$–$6K$, while the top 20% belong to the frequency ranges $>50K$.

6 Experimental framework

For our experiments, we divided both our datasets into different time periods to run our comparison algorithm across these time periods. For the Google books dataset, we created eight DT-based networks for time periods[7]: 1520–1908, 1909–1953, 1954–1972, 1973–1986, 1987–1995, 1996–2001, 2002–2005, and 2006–2008 (Riedl *et al.* 2014). Each time period corresponds to roughly equal-sized data. We will use the symbols T_{g1} to T_{g8} to denote these time periods. Similarly for the Twitter dataset, we created two DT-based networks for time periods: 2012 and 2013. We will use the symbols T_{t1} and T_{t2} to denote these time periods. We then executed our comparison algorithm: (a) within the Google books data to identify the word sense change within a media, and (b) across Google books and Twitter data to identify the word sense change across different media. Since we did not have sufficient Twitter data for this kind of temporal analysis, we could not run comparison within the Twitter data. We found the following parameters for the CW clustering algorithm suitable for our experiments: The size of the neighborhood of a word (N) was set to 200. The edge density inside each of these neighborhoods (n) was set to 200 as well. The parameter for regulating the cluster size was set to option (a) (cf. Section 4)

[7] Available for download at http://sourceforge.net/p/jobimtext/wiki/

Table 3. *Number of candidate birth senses within the Google books data*

	T_{g2}	T_{g3}	T_{g4}	T_{g5}	T_{g6}	T_{g7}	T_{g8}
T_{g1}	2,498	3,319	3,901	4,220	4,238	4,092	3,578
T_{g2}		1,451	2,330	2,789	2,834	2,789	2,468
T_{g3}			917	1,460	1,660	1,827	1,815
T_{g4}				517	769	1,099	1,416
T_{g5}					401	818	1,243
T_{g6}						682	1,107
T_{g7}							609

to favor smaller clusters by downweighing the influence of nodes linearly by their degree[8], see (Biemann 2010) for a detailed account on the influence of parameters. For our comparison algorithm mentioned in Section 5.3, we used the following threshold values. For comparison within the Google books data, we set the value of the constant A_1 to be 30% and A_2 to be 80%. For comparison across the Google books and Twitter data, we set these values as 45% and 90%. The results were quite sensitive to the choice of parameters. For instance, while comparing within the Google books data for the time-periods 1909–1953 and 2002–2005, we obtained fifty-two candidate split/join occurrences and the success rate was 46% for split and 43% for join using WordNet alignment. If we change A_1 to 20% for the same experiment, the success rate decreases to 36% for split and 28% for join and a lot of false positives are obtained. If we change A_1 to 40%, we obtain only thirteen candidate split/join words and thus, many viable candidate words are missing from the result.

6.1 Signals of sense change within the Google books data

Within the Google books data we ran our comparison algorithm between all pairs of time periods (T_{g1} to T_{g8}). It produced twenty-eight candidate word lists. Then we pruned each of these lists though the multistage filtering technique discussed in Section 5.6. Table 3 shows the number of candidate birth senses we got in all of these comparisons. The rows correspond to the earlier periods and the columns correspond to the later periods. Each element in the table corresponds to the number of candidate words flagged due to birth case by comparing the corresponding earlier and later periods.

Table 3 shows a clear trend. For most of the cases, as we go from left to right along a row in the table, the number of candidate birth senses tends to increase. Similarly, this number decreases as we go from top to bottom along a column in the table. If we move along a row from left to right the time interval increases, but if we move along a column from top to bottom the time interval decreases. One can intuitively expect more sense change if the interval increases. In fact, while moving from top to bottom along the diagonal, the candidate words tend to decrease. This

[8] Data available at `http://sf.net/p/jobimtext/wiki/LREC2014_Google_DT/`

Table 4. *Number of candidate birth senses across the Google books and Twitter data*

	T_{t1}	T_{t2}	T_{t12}
T_{g2}	6,143	6,175	2,328
T_{g7}	6,084	6,147	2,325
T_{g8}	6,145	6,204	2,337

corresponds to the fact that the number of year-gaps in each time period decreases as we move downwards, *e.g.* in T_{g1} (1520–1908) there is over three centuries of year-gap, while in T_{g8} (2006–2008) this gap is only two years.

6.2 Signals of sense change across the Google books and Twitter data

For comparing the Google books with Twitter data, we selected three representative time periods (T_{g2}, T_{g7}, and T_{g8}) from the Google books data, then we ran our comparison algorithm between each of them with both the time periods (T_{t1} and T_{t2}) of Twitter data. In each case, after getting the candidate word lists for T_{t1} and T_{t2} we took an intersection of these two lists to get the candidates with a stable sense change across these two Twitter time periods (T_{t12}). Please note that we call a sense change from T_{gi} to T_{t1} 'stable' if the same sense change was also detected while comparing T_{gi} to T_{t2}. Table 4 shows the number of candidate birth senses we obtained in these comparisons. The first two columns correspond to the two Twitter time periods and the third column corresponds to their intersection. The rows correspond to the Google books time periods.

One can observe from Table 4 that the number of candidates for sense change is very high across media in comparison to within a media.

6.3 Stability analysis & sense change location

Formally, we consider a sense change from tv_i to tv_j **stable** if it was also detected while comparing tv_i with the following time periods tv_ks. This number of subsequent time periods, where the same sense change is detected, helps us to determine the **age** of a new sense. Similarly, for a candidate sense change from tv_i to tv_j, we say that the **location** of the sense change is tv_j if and only if that sense change does not get detected by comparing tv_i with any time interval tv_k, intermediate between tv_i and tv_j.

Table 3 indicates a large number of candidate words for sense change, yet not all of these candidates can be considered stable, requiring us to prune them on the basis of a stability analysis. Further, note that results in Table 3 do not indicate the exact time when the change took place: many of the candidate birth senses between T_{g1} and T_{g6} might be contained also in the set of candidate births between T_{g2} and T_{g5}. We prune these lists further based on the stability of the sense, as well as to locate the approximate time interval, in which the sense change might have occurred.

Table 5 shows the number of stable senses obtained during comparisons. For instance, while comparing T_{g1} with T_{g2}, 2,498 candidates were flagged as 'birth'.

Table 5. *Number of candidate birth senses obtained for different time periods*

	T_{g2}	T_{g3}	T_{g4}	T_{g5}	T_{g6}	T_{g7}
T_{g1}	2,498	3,319	3,901	4,220	4,238	4,092
Stable	537	989	1,368	1,627	1,540	1,299
Located	537	754	772	686	420	300
T_{g2}		1,451	2,330	2,789	2,834	2,789
Stable		343	718	938	963	810
Located		343	561	517	357	227

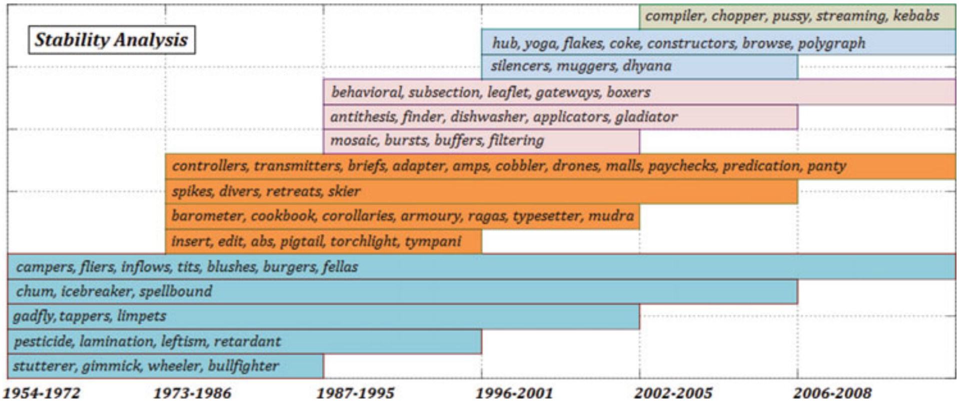

Fig. 5. (Colour online) Examples of birth senses placed on a timeline as per their location as well as age.

However, only 537 of those were stable. What it implies is that while comparing T_{g1} and T_{g3}, only 537 words out of 2,498 were flagged as birth again and thus, were called ‘stable’ birth clusters. Table 5 also shows the number of stable sense changes located in that particular time period. For instance, while comparing T_{g1} and T_{g3}, 989 out of 3,319 candidate birth clusters were stable (*i.e.* also detected while comparing T_{g1} and T_{g4}) and only 754 out of these 989 were located there. What this implies is that other 245 stable senses had already been detected as ‘birth’ in T_{g2} and were therefore, located in T_{g2}. While choosing only the stable senses decreases recall, we found this to be beneficial for the accuracy of the method.

Once we were able to locate the senses as well as to find the age of the senses, we attempted to select some representative words and plotted them on a timeline as per the birth period and their age in Figure 5. The source time period here is 1909–1953. For instance, the entries $\{hub, yoga, flakes, \ldots\}$ in Figure 5 correspond to the fact that while comparing with 1909–1953 sense clusters, the sense changes for these words were first observed in 1996–2001. This sense change was observed during the comparison of 1909–1953 with 2002–2005 and 2006–2008 as well. On the other hand, the sense change for $\{silencers, muggers, \ldots\}$ was first observed in 1996–2001 and also detected in 2002–2005, but was absent while comparing 1909–1953 with 2006–2008.

Below, we give examples of some of the unstable sense changes (birth clusters) obtained by comparing 1909–1953 with 1996–2001. These changes were unstable since these birth clusters were not observed while comparing 1909–1953 with 2002–2005 or later time periods.

- **algebra** - {*grammars, predicates, expressions, formalism, axioms, theorem, calculus, transformation* ...}
- **polarity** - {*antagonism, dichotomies, divide, oscillation, differentiation, distinction, conflict, congruence...*}
- **diamonds** - {*metals, tungsten, graphite, nickel, copper, chrome, uranium, tin, platinum, silver...*}

7 Evaluation framework

Our evaluation strategy is two-fold. First, we compared between the Google books data from two different timestamps; next we did the same comparison between the books data and the Twitter data. Sense changes are classified as either birth (arrival of new sense) or split/join (joining of older senses into one or splitting of older sense into two) or death of a sense. We present a few instances of the resulting clusters in the paper and refer the reader to the supplementary material[9] for the remainder of the results.

7.1 Manual evaluation

Books *versus* **Books comparison:** The split-join algorithm produced good results for all the three cases namely birth, split and join. We randomly selected candidate words from each type (birth, split, and join) and consulted a standard dictionary[10] to check whether the cluster of a candidate word spells out a change in sense. During comparison, 1909–1953 and 2002–2005 were our reference timescales. We randomly selected forty-eight candidate birth words and twenty-one random split/join words for inspection. The accuracy as per manual evaluation was found to be 60% for the birth cases and 57% for the split/join cases.

An interesting side note on this result is that the candidate words can be partitioned into several genres. We found twenty-two technology-related words,

[9] `http://cse.iitkgp.ac.in/resgrp/cnerg/nle2014_wordsense/`

[10] We used New Oxford American Dictionary for manual evaluation, as it contains old as well as new senses of every word. The senses however are not time-stamped. To decide which sense is 'new' or 'old', we consult multiple dictionaries, such as dictionary.reference.com. An example entry in the New Oxford American Dictionary for the word "tripe" is:

(1) the first or second stomach of a cow or other ruminant used as food
(2) informal nonsense; rubbish: you do talk tripe sometimes.

Origin: Middle English: from Old French, of unknown origin.
Corresponding entry in dictionary.reference.com:
c.1300, from Old French tripe "entrails used as food" (13c.), of unknown origin, perhaps via Spanish tripa from Arabic therb "suet" (but also said to mean "fold of a piece of cloth"). Applied contemptuously to persons (1590s), then to anything considered worthless, foolish, or offensive (1892).

Table 6. *Manual evaluation for seven randomly chosen candidate birth clusters from Books 1909–1953 versus Books 2002–2005 comparison*

Sl No.	Candidate word	Birth cluster	Evaluation judgment
1	scroll	*navigate, browse, sort, sift, flip, browse*	**Yes**, New usage related to computers
2	modem	*cables, adapter, devices, subsystem, projector*	**Yes**, New sense related to network
3	caller	*browser, compiler, sender, routers, workstation*	**Yes**, New sense related to 'digital caller'
4	scanner	*ultrasound, images, ct, scanner, imaging*	**Yes**, The new usage related to 'electronic scanner'
5	quiz	*contest, prize, contests, marathon, bowl, games*	**No**, this looks like a false positive
6	select	*cancel, ctrl, menus, panel, query, button, font*	**Yes**, computer related sense
7	pesticide	*pollution, sewage, waste, fertilizer, manure*	**No**, false positive

three words from economics, three slangs, and two general words in the birth sample. In the split-join examples, we got three technical words while the rest of the words were general. So the key observation is that the birth words detected from our algorithm were mainly from the technical fields where the candidate cluster is new, whereas the split-join instances are mostly general.

Table 6 shows the evaluation results for a few candidate words, flagged due to birth. Columns correspond to the candidate words, words obtained in the cluster of each candidate word (we will use the term 'birth cluster' for these words, henceforth), which indicated a new sense, the results of manual evaluation as well as the possible sense this birth cluster denotes. Table 7 shows the corresponding evaluation results for a few candidate words, flagged due to split or join.

Books *versus* **Twitter comparison:** We have applied the same strategy between Books and Twitter data. Table 8 shows the corresponding evaluation results for a few candidate birth words. We randomly selected fifty candidate birth words and got thirty-four true positives, thus achieving a 70% success rate. Among the true positives, twelve correspond to technical words and eleven correspond to slang. When comparing two different media, we did not observe any split or join of senses: senses distributions are different between media to the extent that some senses are missing (or so underrepresented that our method cannot

Table 7. *Manual evaluation for three randomly chosen candidate split/join clusters from Books 1909–1953 versus Books 2002–2005 comparison*

Sl No.	Candidate Word	Source and target clusters
1	mantra **(join)**	S_1: *sutra, stanza, chants, commandments, monologue, litany, verse ...* S_2: *praise, imprecation, benediction, salutation, eulogy ...* T: *spell, sutra, rosary, chants, blessing, prayer ...*
	Yes, the two seemingly distinct senses of mantra - a contextual usage for chanting and prayer (S_1) and another usage in its effect - salutations, benedictions (S_2) have now merged in T.	
2	continuum **(split)**	S: *circumference, ordinate, abscissa, coasts, axis, path, perimeter, arc, plane axis ...* T_1: *roadsides, corridors, frontier, trajectories, coast, shore...* T_2: *arc, ellipse, meridians, equator, axis ....*
	Yes, the split S_1 denotes the usage of 'continuum' with physical objects while the split S_2 corresponds to its usages in mathematics domain.	
3	headmaster **(join)**	S_1: *master, overseer, councillor, chancellor, tutors, captain, general, principal ...* S_2: *mentor, confessor, tutor, founder, rector...* T: *chaplain, commander, surveyor, coordinator, consultant ...*
	No, it seems a false positive	

Table 8. *Manual evaluation for seven randomly chosen candidate birth clusters from Books 2002–2005 versus Twitter 2012–2013 comparison*

Sl No.	Candidate word	Birth cluster	Evaluation judgment
1	mix	*music, vocal, tunes, version, playlist, concert, mixtape*	**Yes**, New usage related to DJing
2	cranberries	*evanescence, fighters, roach, aerosmith, adele*	**Yes**, New usage related to the Irish rock band Cranberries
3	brownie	*chocolate, caramel, toffee, pretzel, brownies*	**Yes**, New sense as small chewy cakelike cookie
4	tripe	*coward, jerks, cretin, prick, pricks*	**Yes**, The new usage related to slang[11]
5	sneakers	*casual, mens, nike, polo, boot*	**Yes**, New meaning related to shoe

detect them) on one media; it is not the case, however, that one media uses a certain word sense in a more differentiated way ('split') than the other.

[11] While the New Oxford American Dictionary lists a similar sense for 'tripe' originating in the year 1892, this sense had apparently fallen out of use on books but re-gained popularity in the social media.

While we have not conducted a full error analysis on the false positives, we noted a pattern that sheds light on possible improvements of the method. The main source of false positives was due to usages *versus* senses – a typical effect when characterizing meaning distributionally, see (Erk, McCarthy and Gaylord 2010): while the clustering seems stable and finds coherent sets of words, they sometimes are grouped due to common contexts and not due to sense distinctions. For example, we found a cluster for 'acknowledgements' corresponding to section and page referrals such as 'seq, pages, iii, xiv, ..' as well as another cluster corresponding to headings such as 'introduction, references, footnotes ..' – both clusters correspond to the sense of 'acknowledgement section', but one of them manifested itself only in the later period for some reason. A possible improvement would identify usage clusters and attempt to cluster them according to their underlying sense distinctions.

7.2 Automated evaluation with WordNet

Apart from manual evaluation, we also designed a few automated evaluation frameworks for the candidate words. For this purpose, we extensively used WordNet[12]. For most of our experiments, we have used WordNet (Fellbaum 1998) version 3.0 (released in December 2006). It contains 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense pairs. The use of a lexical-semantic resource like WordNet in automatic setups for word sense disambiguation can be criticized since lexical resources and word sense induction methods might organize senses differently, yet equally motivated, cf. (Kilgarriff 1997). However, even if quantitative results have to be taken with a grain of salt for this reason, we still feel that automatic evaluation methods are crucial especially when comparing automatic methods for sense change detection, and WordNet is comprehensive enough to support such an evaluation. In our automated evaluation framework, we measure, how many candidates flagged due to birth/split/join actually correspond to a sense change according to WordNet. In the following section, we present this technique.

7.2.1 Accuracy supported by WordNet

The output of our comparison algorithm are candidate words along with one (for birth case) or more (for split/join case) sense clusters. To verify whether each of these candidate clusters signifies a sense change, we need to map the clusters to some sense or synset in WordNet. We developed a mapper that assigns the most likely WordNet ID for given sense clusters. The mapper is a rather straightforward tool with the purpose of enabling an automated evaluation. For a given word, we identify all the WordNet synsets with this word as candidates. Then, we iterate over the cluster members and increase the scores of the WordNet sense ID candidates if one of the words is contained in their synset as a synset member. Finally, the WordNet ID with the highest score is assigned to the cluster. While we have not formally evaluated the mapper, it is able to assign a WordNet ID for about half of

[12] http://wordnet.princeton.edu/

Table 9. *Success rate of candidate birth senses for Books versus Twitter comparison*

Books time period	Twitter time period	Success rate
T_{g2}	T_{t12}	42%
T_{g7}	T_{t12}	47%
T_{g8}	T_{t12}	44%

the clusters, and the large majority of these assignments make sense for clusters of size 5 or larger.

Equipped with an automatic means of mapping cluster senses to WordNet, we present the evaluation technique in the following. We only use data points where we could successfully map all involved clusters to a WordNet sense.

Birth: Each word with a birth cluster (cluster that was absent before) was considered a candidate. To verify that this cluster signifies a new sense, first, we find the sense ID of the birth cluster; then, we retrieve the WordNet sense IDs of all the CW clusters of that word in the earlier period; if all of them are different from the sense ID of the birth cluster, we call it a 'success'; else we call it a 'failure'.

Split: Candidates for split case are words where an earlier single cluster was separated in two or more clusters in a later period. To verify that this signifies a sense change, first, we find the sense IDs of all involved CW clusters mentioned before; then, we check if the sense IDs of the two later clusters are different and one of them is the same as that of the source cluster which we interpret as an indication that a new sense has emerged; if this happens, we call it a 'success'; else we call it a 'failure'.

Join: For the join case, each candidate word is produced with two CW clusters of the earlier period and one CW cluster of the later period, indicating the fact that our algorithm detected that two clusters in the previous period were merged into a single cluster in the later period. To verify that this signifies a sense change, first, we find the sense IDs of all CW clusters involved; then, we check if the sense IDs of the two earlier clusters are different and one of them has the same ID as the later cluster, which signifies that an older sense has vanished; if that happens, we call it a 'success'; else we call it a 'failure'.

As outlined above, we computed the success rates of birth, split and join cases individually for Books *versus* Books comparison. For this, we used the candidate lists obtained by comparing the 1909–1953 data with all the subsequent time periods. Figure 6 shows the distribution of these rates for different cases.

For Books *versus* Twitter comparison, we computed the success rate of only the birth cases. Table 9 shows the success rate assuming three different time periods (T_{g2}, T_{g7} and T_{g8}) for constructing T_{t12}.

After completing these evaluations, we manually verified some of the words flagged as birth that were assessed as success according to WordNet. Along with

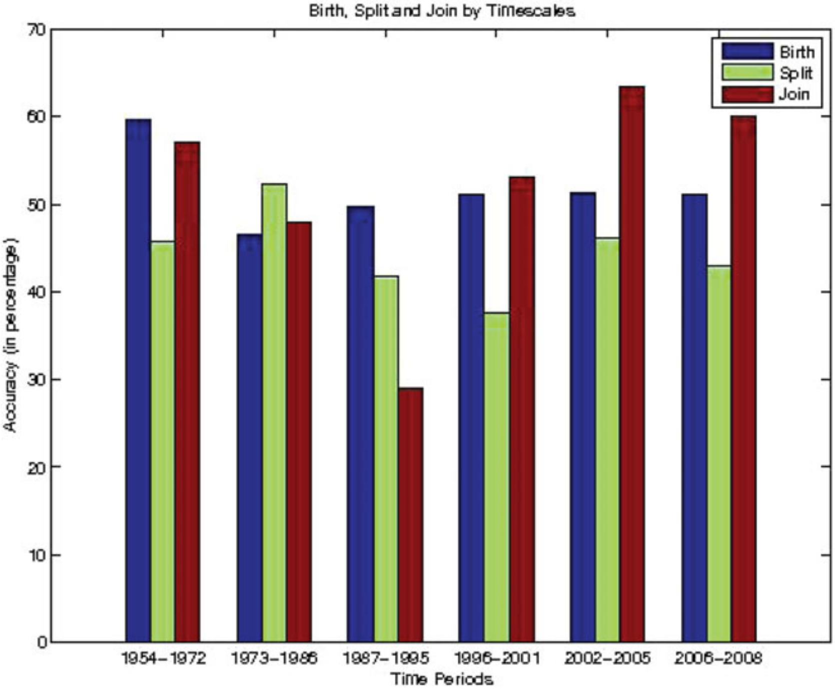

Fig. 6. (Colour online) Distribution of success rates for birth, split and join cases in Books (1909–1953) *versus* Books (subsequent time periods) comparison.

this we also looked into the WordNet senses they were mapped to. Table 10 shows examples where the evaluation identified correct birth clusters.

7.3 Evaluation using slang list

Slangs are words and phrases that are regarded as very informal, and are typically restricted to a particular context. New slang words come up every now and then, and this plays an integral part in the phenomenon of sense change. We therefore, decided to perform an evaluation as to how many slang words were being detected by our candidate birth clusters. We used a list of slangs available from the slangcity website[13]. We collected slangs for the years 2002–2005 and found the intersection with our candidate birth words from 1909–1953 *versus* 2002–2005 comparisons. Note that the website had a large number of multi-word expressions that we did not consider in our study. Further, some of the words appeared as either erroneous or very transient (not existing for more than a few months) entries, which had to be removed from the list. All these removals left us with very little space for comparison; however, despite this we found twenty-five slangs from the website that were present in our birth results, *e.g.* 'bum', 'sissy', 'thug', 'dude', *etc*. For evaluating Google books *versus* Twitter results, we took the candidate birth clusters obtained from T_{g2} *versus* T_{t12}, and found intersection with the slangs up to the year 2008. We

[13] `http://slangcity.com/email_archive/index_2003.htm`

Table 10. *Example of randomly chosen candidate birth clusters, obtained by comparing Twitter (2012) with Books (2002–2005), mapped to WordNet*

Sl No.	Candidate word	Birth cluster	Synset ID, WordNet sense
1	hr	*operations, senior, accounting, customer, assistant, sales, compliance, media, payroll, marketing*	**15227846**, human resources personnel
2	jaguar	*suzuki, dodge, chrysler, honda, chevrolet, ford, chevy, triumph, jeep, peugeot, fiat, cadillac*	**2128925**, a popular car brand
3	villas	*grille, lakes, avenue, inn, suites, pkwy, place, waterfront, leisure, ave, hotel, hills*	**11366405**, related to real estate
4	buffoons	*psychopath, creatures, commentators, statement, bigots, wanker, rhetoric, cretin, morons*	**10100761**, a foolish human being
5	conglomerate	*corporation, companies, firm, manufacturer, firms, business, group, company*	**8059412**, a corporation consisting a number of subsidiary companies
6	starship	*beatles, brothers, halen, styx, browne, mellencamp, revival, band, jovi*	**4304215**, Jefferson Starship, American Rock Band

found seventy-three slangs in this list that were also present in the candidate birth results.

7.4 Evaluation of candidate death clusters

While this paper is mainly concerned with birth of new senses, we also shortly discuss the case where senses get obsolete and move out of the vocabulary. While an in-depth analysis goes beyond the scope of this paper, we selected some interesting candidate 'death' senses. Table 11 shows some of these interesting candidate words, their clusters and their probable original meaning searched by the authors. All of these words are still being used in today's world but their original meanings are more or less lost now.

8 Conclusion

In this paper, we presented a completely unsupervised and automatic method to detect word sense changes by analyzing millions of digitized books archived spanning several centuries as well as millions of tweets posted every day on the social media platform Twitter. In particular, we constructed DT-based networks over eight different time windows for the Google books data and over two different time periods for the Twitter data, clustered these networks and compared these clusters to identify the emergence of novel senses. We then used our split/join based

Table 11. *Some representative examples for candidate death sense clusters*

Sl No.	Candidate word	Death cluster	Vanished meaning
1	sundae	*orchards, plantings, leaves, chips, tree, crop, harvest, plantation, orchard, grove, trees, acreage, groves, plantations, bushes, bark*	Origin: unsure
2	blackmail	*subsidy, rent, presents, tributes, money, fine, bribes, dues, tolls, contributions, contribution, customs, duties ...*	Origin: denoting protection money levied by Scottish chiefs
3	os	*condyle, clavicle, sacrum, pubis, tibia, mandible, vertebra, humerus, patella, maxilla, tuberosity, sternum, femur...*	Origin: a bone in anatomy/zoology
4	phrasing	*contour, outline, construction, handling, grouping, arrangement, structure, modelling, selection, form ...*	in the sense 'style or manner of expression': via late Latin Greek phrases

framework within the Google books data to identify the word sense change within a media, and across Google books and Twitter data to identify the word sense change across different media. The performance of our method has been evaluated manually as well as by an automated evaluation using WordNet and a list of slang words. Through manual evaluation we found that the algorithm could correctly identify 60% birth cases from a set of 48 random samples and 57% split/join cases from a set of twenty-one randomly picked samples within the Google books data. Across the Google books and Twitter data, the algorithm could correctly identify 70% birth cases from a set of fifty samples. We observe that in 51% cases the birth of a novel sense is attested by WordNet for a representative sample within the Google books data. WordNet evaluation also attests that for this sample, in 46% cases a new sense has split off from an older sense and in 63% cases two or more older senses have merged in to form a new sense. Across the Google books and Twitter data, a novel sense was attested for 42–47% of the cases for various samples. These results might have strong lexicographic implications and many of the words detected by our algorithm would be candidate entries in WordNet if they were not already part of it.

Future research directions based on this work are manifold. On one hand, our method can be used by lexicographers in designing new dictionaries where candidate new senses can be semi-automatically detected and included, thus greatly reducing the otherwise required manual effort. This method can be directly used for various NLP/IR applications like semantic search, automatic word sense discovery as well as disambiguation. For semantic search, taking into account the newer senses of the word can increase the relevance of the query result. Similarly, a disambiguation engine informed with the newer senses of a word can increase the efficiency of disambiguation, and recognize senses uncovered by the inventory that would otherwise have to be wrongly assigned to covered senses. To make the method directly applicable in practice without manual intervention, however, it should be made less sensitive to the choice of parameters and its precision needs to be increased.

In addition, this method can also be extended to the 'NNP' part-of-speech (*i.e.* named entities) to identify changes in role of a person/place. Furthermore, it would be interesting to apply this method to languages other than English and to try to align new senses of cognates across languages.

References

Allan, J., Papka, R., and Lavrenko, V. 1998. On-line new event detection and tracking. In *Proceedings of SIGIR*, Melbourne, Australia, pp. 37–45.

Bamman, D., and Crane, G. 2011. Measuring historical word sense variation. In *Proceedings of JCDL*, New York, NY, USA, pp. 1–10.

Biemann, C. 2006. Chinese whispers – an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs*, New York City, NY, USA, pp. 73–80.

Biemann, C. 2010. Co-occurrence cluster features for lexical substitutions in context. In *Proceedings of TextGraphs 5*, Uppsala, Sweden, pp. 55–59.

Biemann, C. 2012. Structure Discovery in Natural Language. In: *Theory and Applications of Natural Language Processing*. Springer, Berlin Heidelberg. ISBN 978-3-642-25922-7.

Biemann, C. 2012. Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources & Evaluation* **47**(1): 97–112. doi 10.1007/s10579-012-9180-5

Biemann, C., and Riedl, M. 2013. Text: now in 2D! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling* **1**(1): 55–95.

Blei, D., and Lafferty, J. 2006. Dynamic topic models. In *Proceedings of ICML*, Pittsburgh, Pennsylvania, pp. 113–120.

Bond, F., Isahara, H., Fujita, S., Uchimoto, K., Kuribayash, T., and Kanzaki, K. 2009. Enhancing the Japanese WordNet. In *Proceedings of Workshop on Asian Language Resources*, Suntec, Singapore, pp. 1–8.

Cook, P., Lau, J. H., Rundell, M., McCarthy, D., and Baldwin, T. 2013. A lexicographic appraisal of an automatic approach for detecting new word senses. In *Proceedings of eLex*, Tallinn, Estonia, pp. 49–65.

Cook, P., and Stevenson, S. 2010. Automatically identifying changes in the semantic orientation of words. In *Proceedings of LREC*, Valletta, Malta, pp. 28–34.

Erk, K., McCarthy, D., and Gaylord, N. 2010. Investigations on word senses and word usages. In *Proceedings of ACL*, Suntec, Singapore, pp. 10–18.

Evert, S. 2005. The statistics of word cooccurrences. Dissertation, Stuttgart University.

Fellbaum, C. (ed.) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Firth, J. R. 1957. A Synopsis of Linguistic Theory, 1933–1955. *Studies in Linguistic Analysis*, Blackwell, Oxford.

Goldberg, Y., and Orwant, J. 2013. A dataset of syntactic-ngrams over time from a very large corpus of English books. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*, Atlanta, GA, USA, pp. 241–247.

Gulordava, K., and Baroni, M. 2011. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *Proceedings of the Workshop on Geometrical Models for Natural Language Semantics*, EMNLP, Edinburgh, UK, pp. 67–71.

Heyer, G., Holz, F., and Teresniak, S. 2009. Change of topics over time – tracking topics by their change of meaning. In *Proceedings of KDIR*, Madeira, Portugal, pp. 223–228.

Ide, N. and Veronis, J. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics* **24**(1): 1–40.

Kilgarriff, A. 1997. I don't believe in word senses. *Computers and the Humanities* **31**(2): 91–113.

Kilgarriff, A., Rychly, P., Smrz, P., and Tugwell, D. 2004. The sketch engine. In *Proceedings of EURALEX*, Lorient, France, pp. 105–116.

Kilgarriff, A., and Tugwell, D. 2001. Word sketch: extraction and display of significant collocations for lexicography. In *Proceedings of COLLOCATION: Computational Extraction, Analysis and Exploitation*, Toulouse, France, pp. 32–38.

Kwong, O. Y. 1998. Aligning WordNet with additional lexical resources. In *Proceedings of the Workshop on Usage of WordNet in Natural Language Processing Systems*, COLING-ACL98, pp. 73–79.

Lin, D. 1997. Zipf's law outside the middle range. *Proceedings of the 6th Meeting on Mathematics of Language*, Florida, USA, pp. 347–356.

Loreto, V., Mukherjee, A., and Tria, F. 2012. On the origin of the hierarchy of color names. *PNAS* **109**(18): 6819–6824.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* **2**: 159–165.

Maity, S. K., Venkat, T. M., and Mukherjee, A. 2012. Opinion formation in time-varying social networks: the case of the naming game. *Phys. Rev. E* **86**: 036110.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., and Aiden, E. L. 2011. Quantitative analysis of culture using millions of digitized books. *Science* **331**(6014): 176–182.

Mihalcea, R., and Nastase, V. 2012. Word epoch disambiguation: finding how words change over time. In *Proceedings of ACL*, Jeju Island, Korea, pp. 259–263.

Mitra, S., Mitra, R., Riedl, M., Biemann, C., Mukherjee, A., and Goyal, P. 2014. That's sick dude!: automatic identification of word sense change across different timescales. In *Proceedings of ACL*, Baltimore, USA, pp. 1020–1029.

Mukherjee, A., Tria, F., Baronchelli, A., Puglisi, A., and Loreto, V. 2011. Aging in language dynamics. *PLoS ONE* **6**(2): e16677.

Navigli, R. 2009. Word sense disambiguation: a survey. *ACM Computing Surveys* **41**(2): 1–69.

Pääkkö, P., and Lindén, K. 2012. Finding a location for a new word in WordNet. In *Proceedings of the Global WordNet Conference*, Matsue, Japan.

Riedl, M., Steuer, R., and Biemann, C. 2014. Distributed distributional similarities of Google books over the centuries. In *Proceedings of LREC*, Reykjavik, Iceland.

Rychlý, P., and Kilgarriff, A. 2007. An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *Proceedings of ACL, Poster and Demo Sessions*, Prague, Czech Republic, pp. 41–44.

Schütze, H. 1998. Automatic word sense discrimination. *Computational Linguistics* **24**(1): 97–123.

Spärk-Jones, K. 1986. *Synonymy and Semantic Classification.* Edinburgh University Press. Edinburgh, Scotland, ISBN 0-85224-517-3.

Tahmasebi, N., Risse, T., and Dietze, S. 2011. Towards automatic language evolution tracking: a study on word sense tracking. In *Proceedings of EvoDyn*, vol. 784, Bonn, Germany.

Wang, X., and McCallum, A. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of KDD*, Philadelphia, PA, USA, pp. 424–433.

Wijaya, D., and Yeniterzi, R. 2011. Understanding semantic change of words over centuries. In *Proceedings of the Workshop on Detecting and Exploiting Cultural Diversity on the Social Web*, Glasgow, Scotland, UK, pp. 35–40.