An Explicit Temporal Change Model for People in the News

Felix Hennig

Master of Science
School of Informatics
University of Edinburgh
2020

Abstract

I present a diachronic change model with novel embeddings for persons explicitly, based on named entity recognition and heuristic name linking. My model improves over a token based baseline model in representing accurate contexts for peoples names. I show that real world events and context changes can be detected in the model using the example of the prime minister of the UK, as well as role changes in the football domain.

All experiments and analysis is conducted on a novel data set, consisting of articles from the British newspaper *The Guardian*, providing a complementing perspective to previous, US-centric analysis of news corpora.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Felix Hennig*)

Acknowledgements

I would like to kindly thank *The Guardian* for providing their API free of charge for scientific usage and as well as the Wikidata project.

I want to thank my supervisor Steve Wilson for his insightful comments and guidance throughout the project.

Table of Contents

Chapter 1

Introduction

Since their popularisation by Mikolov et al. [1], distributed semantic models have not just found applications for downstream natural language processing (NLP) tasks, but have also been used as tools to analyse the corpora that they are trained on. Distributed semantic models encode semantic relationships between the words in the source corpus by embedding the words in a high-dimensional space. By looking at distances between words, word analogies can be found: *Bratwurst* is to *Germany* like *Sushi* is to *Japan*.

But semantics of words are not static over time; language is subject to change. In the analysis of linguistic change over time, news corpora have seen a lot of interest because they contain information about the real world and how relationships between entities change over time. Not only can we look at embeddings of words, but also embeddings of entities such as organisations, people and countries, and their "semantics": what they are associated with over time. For example Kutuzov, Velldal, and Øvrelid [2] traced countries transitioning between war and peace in embedding models trained on large quantities of news articles.

In this work, I build on recent advances in building these temporal embeddings – or *diachronic* embeddings – in combination with an explicit model of people, and explicit embeddings of people. Experiments are conducted on a newly sourced data set of over 2 million news articles by *The Guardian*, a British newspaper.

1.1 Motivation and Problem Statement

Previous work in diachronic embeddings already included the names of people as words that undergo change. For famous people with fairly unique names such as *Barack Obama*, this has already shown context changes over time relating to the real

world, such as *Obama* changing from a university context and his profession as a professor, to becoming a politician and being associated with other politicians from his party [3]. However, treating a name like a normal word leads to some problems. Usually a person is only uniquely identified by their full name, and even parts of a name can refer to different people in different contexts. For example *Obama* is commonly understood to refer to *Barack Obama*, but in an article about his wife the occurrence of the name *Obama* might also refer to *Michelle Obama*. Over time, associations of names to people can also change: *Bill Clinton* and *Hillary Clinton* are an example where in the 2000s the surname would usually refer to *Bill*, but since the 2016 US election campaign, *Clinton* is commonly understood to refer to *Hillary*.

Some names are also generic dictionary words like the surnames of *Theresa May* and *Gordon Brown*. Associating any occurrence of `may` or `brown` to the aforementioned people is problematic.

There are also more complex cases, such as *Mrs. Tony Blair* referring to *Cherie Blair*, but *Mr. Tony Blair* referring to *Tony Blair*. A person can also be mentioned by their role (e.g. “prime minister”) or through a co-reference (“he”, “she”).

Initial success with simple token based methods for certain names has motivated me to tackle these issues and investigate the viability of change modelling techniques for people in general, also including people with common surnames or surnames that happen to be common dictionary words.

1.2 Hypothesis

I hypothesise that **explicit modelling of persons in text allows for meaningful context trajectories for arbitrary people in the news, based purely on text**. Embedding explicit “person-tokens” in the text instead of raw names, which are typically modelled as separate tokens, will allow the tracking of contexts of a person in a text. This will be combined with diachronic models to allow the modelling of context over time. To assess the modelling power of such an approach, generated context trajectories will be compared to real world data.

1.3 Contributions

In this work I present a novel data set of news articles from the British newspaper *The Guardian*, including an overview of the data. Using the data set I provide an

explicit look at the people that are mentioned in the data, how they are distributed – also compared to other non-name words – and I quantify how prevalent the problem of duplicate surnames is. Using the DynamicWord2Vec diachronic embedding model I build a baseline model using word embeddings and a new model embedding full person names. In the experiment analysis I show how the explicit embedding of names improves the representation of persons qualitatively (Section 5.5) and quantitatively (Section 5.6).

Chapter 2

Background & Related Work

This work lies at the intersection of computational social science and natural language processing (NLP). I will explain word embeddings and diachronic models, discuss the commonly used data sets and present related work building on top of them to model real world effects. I will also present some more broadly related work which uses computational methods on text to make claims about the world. Lastly I will take a brief look at modelling people in text.

2.1 Diachronic Word Embeddings

A word embedding is a type of distributional semantic model, meaning that it represents semantics of words based on the distribution of the words in the text. The *diachronic* attribute refers to models that also model a temporal dimension, along which changes in semantics can be observed.

Mikolov et al. [1] popularised neural network based word embeddings with their Word2Vec model. A word embedding model *embeds* words in a vector space by assigning them vectors, indicating their position in the space. The vector space and vectors are typically in a dimensionality of 50 to 300 dimensions. The distances between words in this space model a sense of relatedness or association of words. Distance is usually measured as the *cosine distance* between vectors.

The actual positions are created from an underlying corpus of text in which the words from the vocabulary are observed in context of other words. The models are based on the hypothesis that the semantics of a word can be defined in terms of its context [4]. Mikolov et al. [1] used a neural model to create the embeddings, but the underlying principle is that of *mutual information* [5]: When two words appear

together frequently, the presence of either one provides information about the other. For example `Trump` and `president` are likely to appear together. In the embedding space, all of these binary relationships are encoded in a low-dimensional space. This also clusters words together that might not appear together, but appear together with a third, shared context (e.g. `Trump` and `Bush` share the context `president`).

Word embeddings have seen widespread adoption as word representations in larger NLP systems such as neural machine translation, but have also been used as an immediate tool for the analysis of word relatedness.

In that vein, *diachronic* word embeddings incorporate a temporal dimension and allow the inspection of changes in association over time. Models are usually built by creating multiple discrete embeddings for consecutive, evenly sized time ranges such as years [2, 3, 6], although continuous models exists too [7]. Word embeddings encode relatedness between words in the relative vector positions between the embeddings; when multiple embedding spaces are created for different time slices, each word embedding needs to not only be positioned relative to other words in the same time slice, but also to itself in the adjacent embedding spaces. Various approaches to this problem have been proposed:

Trained embedding spaces can be aligned subsequently, by rotating one embedding space to minimize distance for each word to itself in the other embedding space [6]. This is based on the idea that most words do not change their context. Consecutive embeddings can also be trained sequentially, using embedding n as a starting point for embedding $n+1$, however experiments have shown that this approach is inferior and also does not allow for parallel training [8].

Yao et al. [3] presented a new approach for embedding creation – *DynamicWord2Vec* (DW2V) where the alignment to adjacent time slices already happens in the embedding creation step, removing the need for subsequent aligning. This approach is also more robust to sparse data. In this work I will use this approach.

2.1.1 Temporal Data Sets

To create temporal embeddings, the source corpus for the embeddings needs to have temporal information. Depending on the temporal resolution that should be analysed, a variety of data sources are used. Short range word usage changes have been analysed on Twitter data [8] while long term changes over multiple decades to centuries have been analysed in books [9, 6, 10].

For medium range resolution of years, news corpora have seen widespread use [2, 3, 11]. There are academic corpora covering only a singe newspaper [12] or aggregating multiple newspapers [13]. Some researchers also create their own corpora from news content online [3].

The New York Times has been used frequently, either through their published data set *The New York Times Corpus* [12] or by crawling articles from the web [3]. The Gigaword Corpus [13] aggregates 8 news sources, but three of them are American newspapers, including the biggest one which is again the New York Times.

2.2 Computational Social Science

The digitisation of large amounts of media has allowed social scientists to analyse culture and real world developments statistically, through the quantitative analysis of text. This type of research is referred to as *Computational Social Science*.

Michel et al. [9] created a corpus of over 5 million books spanning 200 years to analyse the change of language and culture by analysing word frequencies. They show that inventions such as the telephone or the discovery of DNA is also reflected in text, by the introduction of new words and changes in word frequency.

Garg et al. [10] go beyond frequency analysis and use word embeddings to analyse associations with of jobs with certain ethnic groups and genders. They show that the associations encoded in 100 years of news and books correlate with the actual occupation ratios for genders in specific jobs in the US, showing that text also encodes subtle real world relationships. The analysis uses 10 year chunks of data.

At a more fine grained resolution Zhang et al. [14] use news articles from the New York Times (NYT) to show temporal analogies between the 1990s and the 2000s, such as “iPod” being the 2000s equivalent of a “Walkman” in the 1990s. Yao et al. [3] also use NYT articles and show meaning changes for the words “amazon” and “apple”. In both works, peoples names are analysed alongside other words. Traces for individual people are shown, as well as the association of a role with different names over time (president of the USA, mayor of New York), showing that yearly change trajectories can be generated from news data.

Kutuzov, Velldal, and Øvrelid [2] do event detection instead of gradual context changes, by tracking country names in the news over time and observing state changes between war and peace. They propose the aggregation of multiple words into “concept embeddings” and manage to improve the event detection score significantly this way.

This is an example of a move away from strict word embeddings towards more high-level embeddings.

2.3 Person Detection

Previous analysis of people in corpora as mentioned above is strictly based on the simple mapping of a token to a person, usually the persons surname is used [11, 3]. Previous work already identifies the problem that a word cannot be strictly associated to an individual, due to multiple people sharing a surname [3].

Identifying the occurrence of the name of a person in a text is part of the task of *named entity recognition* (NER). NER identifies the occurrence of a proper noun referring to a person, place or organisation, which is a good starting point to identify mentions of people in text. The latest models are based on neural networks, and while performance is already quite good, the F1 score is still in the range from 85 to 90 [15, 16].

Additionally, there are many more references to a person in a text beyond mentions of their name. A person can be mentioned by their role (“the prime minister”) or with a pronoun as a co-reference (“she”), these co-references can also form chains [17].

Chapter 3

Data Set

For this work I am using a novel data set sourced from the British newspaper *The Guardian*. *The Guardian* provides all their content via an API called OpenPlatform[1], launched in 2009 [18]. This data source has seen only tangential use in the scientific community [19, 20, 21] and has not been used for diachronic models before. It offers interesting properties for temporal embedding research:

The API serves more than 2 million articles, with the majority of articles being published from 2000 onward, allowing the coverage of two decades with high density. The API allows retrieval of articles as soon as they are published, which means that any research can always build on the most recent data, compared to frozen academic corpora.

Compared to other data sets described in Section 2.1.1 which are more America-centric, this data set is sourced from a British newspaper, which also allows to recreate some content analysis that was mostly conduced for American news in a British context. For example, analysing the change of the British prime minister instead of the American president [11, 3].

3.1 Data Retrieval & Processing

The API allows retrieval of full article texts as well as meta data such as publishing date, author, section and tags in `json` for a given date range. I retrieved all information that was retrievable for all articles published from 1995 to 2019, over multiple days in March 2020. I ended up using the two full decades starting in 2000 due to the amount of articles per year being significantly less before 2000. The final data set is described

[1] https://open-platform.theguardian.com/

in more detail in Section 3.2.

Figure 3.1 shows an example of a json representation of a document. The object contains meta information such as publication date, section and associated tags. In the fields of the object the various components of the article are found: headline, trail text, body and byline. Every article has an `id`, which uniquely identifies the article in the API. I also used this ID to make sure the data contained no duplicates. The articles also have a `type`, including textual and non-textual documents. I used documents of the type `article` and `liveblog`. Other categories included interactive content, slideshows, crossword puzzles and audio and video content.

To actually retrieve the text for the document, I used the `bodyText` field, which already contains the body of the article, stripped from HTML tags. I did not include the trail text, which was typically similar to the beginning of the article. I also did not include article titles.

To turn the text into tokens I used the standard tokeniser in spaCy[2]. The resulting data used for training was a collection of token sequences, divided into yearly chunks.

In addition to the publishing date I also kept the section in which the article was published.

3.2 Data Description

The data set contains a total of $2,021,947$ articles containing 1.65 billion tokens. Figure 3.2 shows the distribution of articles over the years, as well as over sections. Each year contains about $100,000$ articles, which is about 270 articles every day. Every document is published in exactly one section, the figure shows the 20 biggest sections across the data set, as well as a "rest" category.

The sections give an insight into the content of the news. World news is the biggest section, there is a lot of reporting on news outside of the UK. Additionally there are dedicated sections for specific countries, such as "US News" and "Australia News". For domestic news, articles are split into content based sections such as "politics", "business", "media" and "sport". It is notable that "football" is a dedicated section, on par with the more generic "sport" section.

Not shown in the plot is the amount of articles per section over time. The "Media" section has fallen in article counts since 2008, and more specific sections have become more used such as "Books", "TV and Radio" or "Film".

[2]https://spacy.io/

```
{"id": "world/2019/sep/01/it-gave-me-hope-[...]",
 "webTitle": "'It gave me hope': New Zealand [...]",
 "type": "article", "sectionId": "world", "sectionName": "World news",
 "webPublicationDate": "2019-08-31T23:30:01Z",
 "webUrl": "https://[...]", "apiUrl": "https://[...]",
 "fields": {
   "headline": "'It gave me hope': New Zealand [...]",
   "standfirst": "<p>Bike riding courses [...]</p>",
   "trailText": "Bike riding courses [...]",
   "main": "<figure [...]> <img [...] /> <figcaption> [...] </figcaption>
      </figure>",
   "body": "<p>Leila Rahimi was so scared [...]</p>",
   "bodyText": "Leila Rahimi was so scared [...]",
   "byline": "Eleanor Ainge Roy in Dunedin",
   "bylineHtml": "<a href=\"profile/eleanor-ainge-roy\">Eleanor Ainge
      Roy</a> in Dunedin"
   "wordcount": "438", "charCount": "2498",
   "firstPublicationDate": "2019-08-31T23:30:01Z",
   "lastModified": "2019-09-01T21:52:08Z",
   "isInappropriateForSponsorship": "false",
   "productionOffice": "AUS",
   "thumbnail": "https://[...].jpg",
   "legallySensitive": "false",
   [...]
 },
 "tags": [{"id": "world/newzealand", "webTitle": "New Zealand",
          "type": "keyword",
          "sectionId": "world", "sectionName": "World news",
          "webUrl": "[...]", "apiUrl": "[...]", "references": []},
         {"id": "lifeandstyle/cycling", [...]},
         [...]],
 [...]}
```


Figure 3.1: A shortened example of an API object representation of a document. Due to space constraints, many keys have been left out. The article text was retrieved from the `bodyText` field.

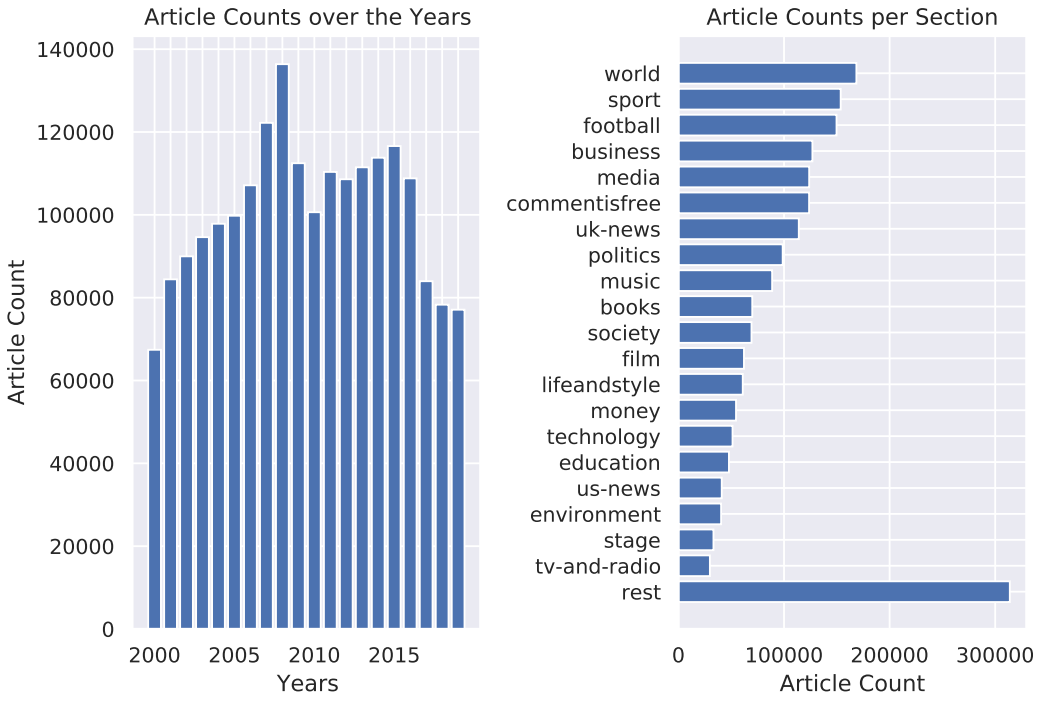

Figure 3.2: The figure on the left shows yearly article counts, the figure on the right shows the article counts per section. The 20 biggest sections are shown, and the rest of the sections are aggregated as “rest”.

The “rest” section aggregates all the smaller sections, which make up a large amount of articles. There are smaller topical sections such as “games”, “law” and “sustainability”. There are also very specific series of articles such as “Women in Leadership” or “Guardian Masterclasses”. Also regional content about specific cities such as “Leeds”, “Cardiff” or “Edinburgh”.

There is also a long tail of sections with less than 10 articles that are very specific.

Chapter 4

Methodology

Below I am describing the approach to extracting references to people in the text and linking them together, as well as the subsequent building of diachronic models.

4.1 Modelling People in the Text

To allow the explicit modelling of people in the text, mentions of people have to be detected and linked to a person. The most distinctive way a person is mentioned in a newspaper text is typically through their surname. This is because people are either addressed by their full name – which includes the surname – or with a title and surname, or more recently also with just their surname.

The problems with this approach have already been mentioned. Surnames are not unique and names can also be dictionary words, which means that the context of usage is important.

Once references to a person have been identified, they need to be linked together. Again the baseline is exact match of tokens or names. But as described earlier, a person can sometimes be referred to by their full name, just a surname or a surname and a title. All these different surface forms refer to the same person.

My approaches to identifying references and the subsequent linking are discussed below. Once a set of mentions have been linked to a person, a new text corpus is built where any span of tokens that is part of a mention is replaced by a pseudo-token representing the person.

4.1.1 Identifying Mentions

To go beyond simple token based name associations, I used a named entity recognition (NER) model to identify the occurrence of names in the text. The Python library *spaCy*[1] provides a neural NER model with an architecture based on Strubell et al. [16]. I used the default pretrained model for English: `en_core_web_sm`. The library creators report an accuracy of around 85% [2] for all entities. I only use the detection of persons, no individual accuracy score is given. The quality of the person detection on the data set is discussed in Section 5.1.

Post-processing: The model includes possessive "'s" in the detected reference of people, these have been removed. The model also includes titles such as "Mr." or "Baroness" in the token span of the person reference. While at first it seems that *Mr. Tony Blair* can just be shortened to *Tony Blair*, there are also mentions of *Mrs. Tony Blair*. Linking this name to *Tony Blair* is wrong, because in this case the expression refers to *Cherie Blair*, *Tony Blair's* wife.

4.1.2 Merging Mentions

After mentions of a person have been identified, mentions of the *same* person need to be identified as such. To solve this problem, I used the division of the data into documents, as well as the temporal nature of the data to link more leniently within articles and also try and merge based on time-local frequency of various names.

Within article linking: In text in general and news articles in particular, it is common to first introduce a person by their full name and subsequently only refer to them by part of their name, in the news this is usually their surname in more formal writing and the first name in more casual writing. So while in general the name *Johnson* is ambiguous, in an article that already mentioned *Boris Johnson* the association of the name *Johnson* with *Boris Johnson* is already salient. So within an article any detected name that only consists of a single token is linked to a previous mention that includes this token.

Across article linking: Across articles I first link based on exact string match of names: Every time *Tony Blair* is mentioned, I assume it is referring to the same person. In combination with the previously mentioned within-article-linking, this is already an effective strategy.

[1] `https://spacy.io/`

[2] `https://spacy.io/usage/facts-figures#spacy-models`

The next step is to link single token names. If a person is very famous, articles mentioning them do not usually introduce them with their full name anymore. This is particularly noticeable with presidents or prime ministers. Before victory at the election they are often referred to by their full name, but afterwards their surname is so commonly used to refer to them, that many news articles will also only use the surname. With the previously described linking steps, these single token surname references are not linked to anything. To merge these occurrences, for every month, every name that consists only of a single token is linked to the person that contains that token and has the most mentions in that month and the previous month. The same is done for names that start with *Mr* or *Mrs*. This helps mapping *Mr Johnson* and *Johnson* to *Boris Johnson*. This also produces a few false positives, such as *Mrs Blair* getting mapped to *Tony Blair*. Doing it month by month allows *Clinton* to be mapped to *Bill Clinton* in one time period and *Hillary Clinton* in another.

This technique also mimics how readers perceive names, a name does not have to be mentioned in full if it is salient in the readers mind, i.e. if the person is well known. If the person is well known, it is likely that they have been talked about a lot in the recent past.

4.2 Diachronic Model

To create diachronic embeddings of persons as well as words I use *DynamicWord2Vec* (DW2V) developed and first described by Yao et al. [3]. A key benefit of this method over other methods is that embeddings for all time slices are trained concurrently, without subsequent alignment. The alignment is done during training of the embeddings, this also creates more stable embeddings.

First, a fixed vocabulary V is defined which is used across all years. The embeddings are not created directly from text with a neural network, as it is done with Word2Vec. The embeddings are calculated based on positive pointwise mutual information (PPMI) between words, which needs to be extracted from the text first. The word embeddings are created in such a way that for any word the cosine distance to other words in embedding space throughout the years closely matches the PPMI values, while also having a high self similarity across the years. The balance between local precision within a year, and alignment across years can be adjusted with a hyperparameter in the training. Levy and Goldberg [5] show that the Word2Vec model and PPMI matrix factorisations are equivalent.

4.2.1 PPMI Matrices

The embeddings are trained on PPMI values. For any two words w_1 and w_2 in V and every time slice t, the PPMI score needs to be calculated. Pointwise mutual information is a measure to indicate how much information one word holds about the other. This is calculated by looking at the ratio between co-occurrences of w_1 and w_2 and individual occurrences of the words. If they occur together more often than it would happen by chance, they have positive mutual information, otherwise they have negative mutual information. It is also possible that the number of co-occurrences is the number expected from random chance, in this case the two words have no mutual information.

Co-occurrence in a text is typically defined as appearing together within k words of each other. Any words within k words before or k after a word w_1 are counted as co-occurring with w_1 for this appearance of w_1.

$$pmi(w_1, w_2) = log\Big(\frac{|V| \cdot c(w_1, w_2)}{c(w_1) \cdot c(w_2)}\Big) \tag{4.1}$$

Equation 4.1 shows how the PMI score for two words can be calculated from word counts or co-occurrence counts respectively; $c()$ gives the count of words or co-occurrences. To get the PPMI score, the values are floored at zero:

$$ppmi(w_1, w_2) = max(pmi(w_1, w_2), 0) \tag{4.2}$$

These scores need to be calculated for every pair of words in V, and so the resulting scores can be stored in a $|V| \times |V|$ sized matrix. Note that mutual information and therefore also the matrix is symmetric. Many scores will be zero, which means the matrix will be quite sparse; this fact allows for small storage of the matrices and also makes it easier to keep them in memory for computation later on.

4.2.2 Creating Embeddings

The embeddings are created based on the PPMI matrices by solving a matrix factorisation problem. Every constraint is encoded mathematically, creating a loss function that needs to be minimised:

$$\begin{aligned} \min_{U(1),\ldots,U(T)} \frac{1}{2}\sum_{t=1}^{T} ||Y(t) - U(t)U(t)^T||_F^2 + \frac{\lambda}{2}\sum_{t=1}^{T} ||U(t)||_F^2 \\ + \frac{\tau}{2}\sum_{t=2}^{T} ||U(t-1) - U(t)||_F^2 \end{aligned} \tag{4.3}$$

The first term ensures that for every year, the embedding models the PPMI values. The second term weighted by λ is a regularising term to prevent the embeddings from getting too large. The third term weighted by τ ensures that adjacent embeddings are aligned with each other.

To actually compute these matrices $U(1)$ to $U(T)$, every time step is aligned on its own. This is the repeated a few epochs, until the embeddings have converged.

$$\begin{aligned}\min_{U(t)} \frac{1}{2}||Y(t) - U(t)U(t)^T||_F^2 + \frac{\lambda}{2}||U(t)||_F^2 \\ + \frac{\tau}{2}(||U(t-1) - U(t)||_F^2 + ||U(t) - U(t+1)||_F^2)\end{aligned} \tag{4.4}$$

Equation 4.5 shows the formula used to minimize the embeddings of a single time slice. We then also introduce a "context" embedding for each word, to break the symmetry of the equation:

$$\begin{aligned}\min_{U(t)} \frac{1}{2}||Y(t) - U(t)W(t)^T||_F^2 + \frac{\gamma}{2}||U(t) - W(t)||_F^2 \\ + \frac{\lambda}{2}||U(t)||_F^2 + \frac{\tau}{2}(||U(t-1) - U(t)||_F^2 + ||U(t) - U(t+1)||_F^2) \\ + \frac{\lambda}{2}||W(t)||_F^2 + \frac{\tau}{2}(||W(t-1) - W(t)||_F^2 + ||W(t) - W(t+1)||_F^2)\end{aligned} \tag{4.5}$$

The same constraints are now enforced for both U and W. Additionally, a new constraint enforces both embeddings to be similar to each other, this constraint is weighted with the hyper-parameter γ. Taking the derivative gives us a formula for simple least squares optimisation with $U(t)A = B$:

$$\begin{aligned}A &= W(t)^T W(t) + (\gamma + \lambda + 2\tau)I \\ B &= Y(t)W(t) + \gamma W(t) + \tau(U(t-1) + U(t+1))\end{aligned} \tag{4.6}$$

This formula is symmetric for W. It is important to note that the formulas change slightly at the edges, because alignment is then only required with a single neighbouring time slice instead of two slices. This means that for both A and B, one of the τ terms is removed.

Once the training is finished, for every time slice there are two embedding matrices U and W. I chose to concatenate them for the final word embeddings.

4.2.3 Implementation

The equation above is already written to optimise a single time slice while keeping the other slices fixed. But $Y(t)$ is a large matrix, although sparse. But if multiplied, it

becomes a large dense matrix. However, the calculation can be done in slices, so only b words are adjusted at the same time, keeping the others fixed.

Both the batches as well as the order in which the time slices are updated are randomised, to prevent any skewed embeddings that would be created by a fixed order. The randomisation of batches is new in my implementation compared to the reference implementation by Yao et al. [3].

4.2.4 Summary

To summarise, first a vocabulary needs to be fixed, in my case I included every word that had at least a minimum count m across the whole time range. Then, PPMI matrices need to be calculated using a window size w. Then a model is trained, with batch size b, n epochs and hyper parameter λ for the regulariser weight, γ for the U, W similarity enforcement and τ for the cross-time alignment. The result of the training is a sequence of embeddings of length T, with one embedding per word in V for every time slice.

Chapter 5

Experiments & Evaluation

In the experiments, the two main questions based on the hypothesis as stated in Section 1.2 are:

1. How well can the explicit person model generate traces for people, compared to a token based baseline?

2. How well do the traces model context changes in the real world?

I will look at this quantitatively as well as through individual examples. Before looking into the diachronic change models, I want to look at the persons in the data and quantify the issue of duplicate and ambiguous surnames. All the results will be discussed in Section 5.7.

5.1 Persons in the Data Set

Before the change models are built it is useful to look into the distribution of people in the text. This makes it more clear what the experiments are built on. I will also quantify issues mentioned in the introduction of this work, about the difficulties of mapping tokens to individual people.

The statistics below were conducted on persons detected with named entity recognition and then basic within-article merging was conducted as described in Section 4.1 and across-article merging was done with only exact matches to avoid skewing the number of mentions for the most mentioned individuals.

There are 2,725,110 persons with 29,943,111 mentions. Table 5.1a shows how many people have how many mentions. The distribution is logarithmic; like vocabularies of corpora, the people in the corpus follow Zipf's Law. A small set of people

Mentions	Person Count
> 5	458,158
> 50	61,828
> 500	7,559
$> 5{,}000$	417
$> 50{,}000$	10

(a) The table shows the number of people that have more than a specific number of mentions. It shows the logarithmic relation.

Person	Count
Tony Blair	144,061
Donald Trump	143,149
David Cameron	119,176
Gordon Brown	116,895
Boris Johnson	84,744
Hillary Clinton	67,700
Chelsea	62,336
Commons	58,079
George Bush	58,018
George Osborne	56,646

(b) The 10 people with more than 50,000 mentions.

Table 5.1: The distribution of people and mentions, as well as the top 10 people by mention count.

is mentioned frequently, and many people are only referred to less than five times (See Table 5.1a). Table 5.1b shows the 10 people that have at least 50,000 mentions. *Chelsea* and *Commons* are false positives, the quality of the name detection is discussed below in Section 5.2. The other 8 people are all politicians, four UK politicians and 4 from the US.

The top 100 mentions are still dominated by politicians, but athletes, in particular football players, make up a large portion too.

The raw frequencies of mentions can be analysed to make approximate inferences about the world. Figure 5.1 shows the mentions of the prime ministers of the UK for the past two decades. The mentions show popularity trends corresponding to their terms as prime ministers. For *Tony Blair*, *Gordon Brown* and *David Cameron* these curves give a good overview of when they were in office, but after 2016 it is not entirely clear who would be prime minister; from the frequencies it does not look as if Theresa May would ever be prime minister. I analysed the prime ministers in the diachronic models in Section 5.5.2.

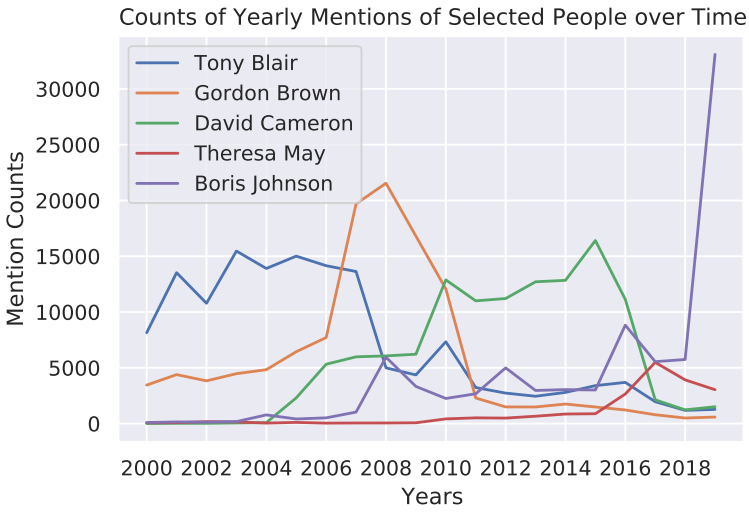

Figure 5.1: The mentions for the 5 prime ministers of the UK from 2000 to 2019.

5.2 Quality of NER

As mentioned in Section 4.1, the NER used has a reported accuracy of around 85%. "Facebook", "Twitter" and "Brexit" were falsely identified as persons, just like the previously mentioned "Commons" – likely detected as a name in the phrase "House of Commons", a British political institution. Besides the already reported "Chelsea", there are also "Tottenham", "Manchester United", "Fulham" and many other sports teams in the false positives. All of these are at least merged together and can still provide useful context.

There were also many instances of false positive detected names of the form "Manchester 1 - 1 Norwich", a match result. In this case, a detected mention like this cannot be merged and "ties up" the words "Manchester" and "Norwich", preventing them from contributing to any embeddings. I did create another model with these false positives removed, but I did not find significant improvements over other models.

5.3 Duplicate and Ambiguous Names

In Section 1.1 I described the problem of mapping tokens directly to people; the two main problems are duplicate names and names that are also dictionary words (*May*, *Swift*, *Stone*). Before going into the experiments I will quantify those phenomena.

First I assess the prevalence of duplicates. I evaluate every year individually, because duplicate names are most problematic if they appear within the same time frame.

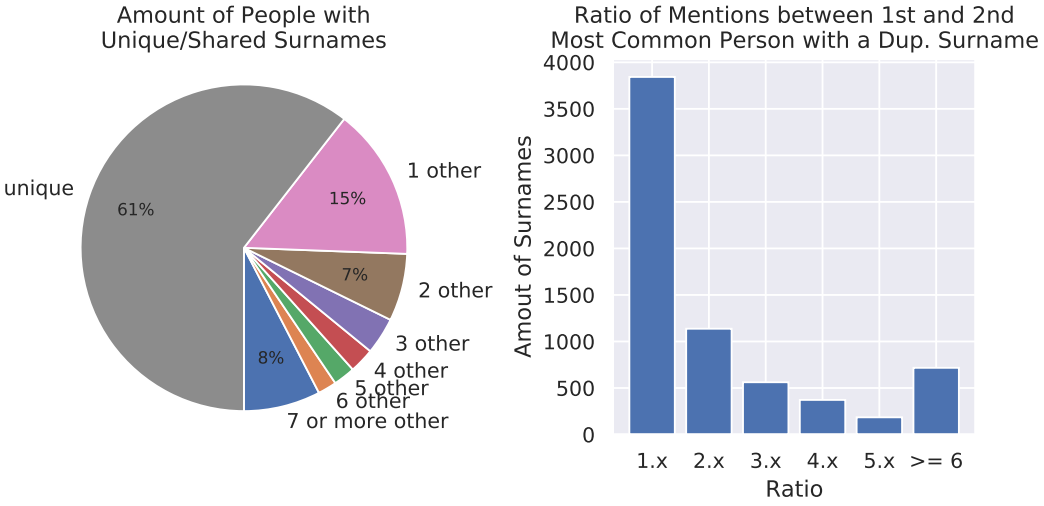

Figure 5.2: The plot on the left shows how common a unique or a shared surname is. For each year, every person with at least 50 mentions was considered. Only other names within the same year were used to identify duplicates. The plot on the right shows the relative frequency of people that share a surname. I.e. if two people share the name *Johnson*, how many times more is the more common *Johnson* mentioned compared to the other?

For every year, I retrieved any person that has at least 50 mentions. Then I grouped them by surname and counted how many people had a unique surname or a surname that they shared with one other person, two others, three others and so on. For people with the same surname I was also interested in whether they would all be mentioned approximately the same amount or if it is common for one person to have many more mentions than the others.

Figure 5.2 shows the results of this analysis. We can see that more than a third of people do not have a unique surname. Keep in mind that only people with at least 50 mentions were considered, if the threshold was lower, the ratio of people with unique surnames would go down. Looking at the ratio of mentions between the most mentioned person with a surname and the second most, we can see that it is not uncommon for a single person to "dominate" a surname, dwarfing any mentions of other people with the same surname. Although in most cases, the first and second most commonly mentioned person with a shared surname do not have a big difference in counts of mentions. Both phenomena have different effects on the embeddings.

If there is a large discrepancy in amount of mentions, the person mentioned more often will "dominate" the embedding for that surname. For example, in 2019 there

are 20 people with the surname *Johnson* that have at least 50 mentions. The most mentioned person is *Boris Johnson*, mentioned 80 times more than the second most mentioned, the golfer *Dustin Johnson*. In a purely token based model, most contexts of *Johnson* will be from *Boris Johnson*. It will be a reasonable embedding for him, but *Dustin Johnson* is hidden.

If a surname is about equally likely to be used for two or more different people, the embedding for that surname will instead be a an even mixture of them, but not really represent anyone properly.

The other phenomenon is that of surnames being also used as normal dictionary words. Due to the false positives in the NER, it is difficult to quantify how many people there are that also have a surname that happens to be a word. In 2019, from all people with at least 200 mentions, *Philip Green*, *Arron Banks* and *Fiona Hill* have the surnames that are most likely to also be used as a regular word. In all three cases, their surnames is 50 times more likely to be used as a regular word than as their surname.

In token based models, both phenomenons – duplicate and ambiguous surnames – can lead to a person being "invisible", because any part of their name is too common on its own. *Taylor* is one of the most common surnames, and the adjective `swift` is also very prevalent. But both words together as the name *Taylor Swift* is much more unique. I am looking at *Taylor Swift* in the embeddings specifically in Section 5.5.1

5.4 Model Setup

After this analysis of the people in the data, the diachronic models will be inspected. Below the parameters of the trained models are described.

The first model I built uses no person detection and is purely token based. This is subsequently called the `token` model, it is the baseline I compare my own model to. My model – subsequently called the `person` model – embeds detected persons using within-article and cross-article merging as described in Section 4.1.

Both models use yearly slices, covering 20 years from 2000 to 2019 (inclusive). To define the vocabulary that is used, only words or names with more than $c_{min} = 500$ occurrences across the whole time span were considered. The vocabularies of both models differ, but both contain around 55,000 types. Many common words are in both vocabularies, most differences are in the names, which appear as individual tokens in the `token` model and as compound names in the *model*.

For the DW2V model Yao et al. [3] provide their hyper-parameter settings as a

Year	`person` model: *Taylor Swift*	`token` model: `taylor`	`token` model: `swift`
2008	Shakira	anderson	timely
2009	Whitney Houston	wright	timely
2010	Iggy Pop	adam	immediate
2011	Selena Gomez	davies	swiftly
2012	Patti Smith	adam	speedy
2013	Beyoncé	craig	timing
2014	Lily Allen	jones	kim
2015	Madonna	smith	adele
2016	Justin Bieber	adam	beyoncé
2017	Beyoncé	smith	recall
2018	Beyoncé	mitchell	swiftly
2019	Madonna	ross	drake

Table 5.2: A comparison of the neighboring tokens/names for *Taylor Swift* over the years. *Taylor Swift* first appeared in the data in 2008. The second column shows the closest names in the person model, the third and fourth shows the closest words to the tokens `taylor` and `swift` in the `token` model individually.

starting point: $\lambda = 10, \tau = \gamma = 50$ and window size 5 for the PPMI matrices, 5 epochs of training. The absence of gold standard embeddings or other high-quality target data hinders a systematic hyper-parameter search. I briefly trained some models using slightly varied parameters to adjust to the different data size, but no large improvements could be found.

5.5 Qualitative Analysis

I am comparing the two models using examples of individual people, analysing the representation of changes for the role of prime minister and inspecting samples of detected changes in the model.

5.5.1 Taylor Swift, Boris and Dustin Johnson

Following up on the claims made in Section 3.2, given the embedding models we can take a look at the American singer-songwriter *Taylor Swift*.

Taylor Swift first appeared in the guardian in 2008, Table 5.2 shows the embedding

Year	token: johnson	person: *Boris Johnson*	person: *Dustin Johnson*
2010	davies	mayor	Francesco Molinari
2011	alex	Bravo Boris	mcilroy
2012	ryan	Ed Miliband	Charl Schwartzel
2013	nick	Iain Duncan Smith	Zach Johnson
2014	joe	Iain Duncan Smith	Jim Furyk
2015	adam	vince	Jason Day
2016	tony	David Cameron	Jason Day
2017	cameron	Theresa May	Jordan Spieth
2018	jeremy	Jeremy Corbyn	Brooks Koepka
2019	boris	Jeremy Corbyn	Phil Mickelson

Table 5.3: The table shows the closest token/person for the token johnson in the token model, and for the names *Boris Johnson* and *Dustin Johnson* in the person model over the last decade of the data.

contexts for the person and token models. We can see that in the person model, the neighbours throughout the years are all musicians, most of them female. When looking for taylor and swift as tokens in the token model, the contexts are not as meaningful. For taylor, the neighboring tokens are simply other common first and last names. For swift there are other words related to the meaning "happening quickly or without delay", such as timely, immediate or speedy. There are however also mentions of adele, beyoncé and drake, all of them are pop musicians.

This discovery is in line with the hypothesised effects above: The common name *Taylor* and the usage of swift as an adjective "dilute" the embeddings for *Taylor Swift*.

Table 5.3 shows the other phenomenon, a shared surname. The politician Boris Johnson shares his surname with the golfer Dustin Johnson. The first column in the table shows the closest token to the token johnson over the years. All tokens are names, but until 2017 they seem to be just common names. From 2017 to 2019 the names are cameron, jeremy and boris, most likely referring to *David Cameron*, *Jeremy Corbyn* and *Boris Johnson's* own first name.

The explicit person embeddings are more informative, putting *Boris Johnson* close to other politicians throughout the years. In 2010, the closest word to him is "mayor"; from 2008 to 2016 *Boris Johnson* was mayor of London.

While the embedding for johnson is influenced by *Boris Johnson* because of the

large amount of mentions of him, *Dustin Johnson* has no effect on the embedding, he has only about a tenth of the mentions of *Boris*. In the explicit name embedding, all the closest names to him are other golf players. With his low mention count, some of the associations can even be attributed to specific events. In 2017 the closest name is *Jordan Spieth*, he came in second after *Dustin Johnson* in the tournament "The Northern Trust". In 2018 *Dustin Johnson* is mentioned a significant amount of times together with *Brooks Koepka* due to an alleged altercation after tensions they had on the evening after a match.

5.5.2 Prime Ministers of the UK

As an example of how well the model represents factual information, we can look at the role of prime minister. To define a vector for the role, the vector of the incumbent prime minister in 2010, the middle of the time range, is taken. This is is *David Cameron*. Table 5.4 shows the closest person in the `person` model, as well as the closest token in the `token` model. The `person` model is mistaken only twice, in 2004 and 2015, while the `token` model is wrong 8 times. Furthermore, the token model does not allow to just look for people and it is difficult to know how to associate tokens with names.

The biggest differences are seen for *Gordon Brown*, *Theresa May* and *Boris Johnson*. The `token` model does not retrieve *May* or *Johnson* at all, and retrieves *Gordon Brown* only one out of 3 years, and with his first instead of last name. Looking into the models, *Brown* and *May* are are not associated with persons but instead with other dictionary words; `brown` is associated with other colours, and `may` is associated with words such as `could` or `might`. *Boris Johnson's* first name is reasonably unique but his last name is quite common.

While the first names show trajectories that are reasonably informative, the last names do not. A lot of reporting on famous people is done with just their last name though. Linking their last name to their first name therefore improves the association of names for these people.

5.5.3 Largest Change Spikes

After looking at an example of how well real world change is recalled in the model, this section analyses if sampled detected changes are indicative of actual real world context changes.

The largest context changes in the model were identified by calculating cosine

Year	Name	Token
2000	Tony Blair	hague (blair)
2001	Tony Blair	chancellor (blair)
2002	Tony Blair	blair
2003	Tony Blair	blair
2004	Michael Howard*	chancellor (blair)
2005	Tony Blair	chancellor (gordon)*
2006	Tony Blair	chancellor (gordon)*
2007	Gordon Brown	blair*
2008	Gordon Brown	gordon
2009	Gordon Brown	cameron*
2010	David Cameron	cameron
2011	David Cameron	cameron
2012	David Cameron	cameron
2013	David Cameron	cameron
2014	David Cameron	cameron
2015	Jeremy Corbyn*	cameron
2016	Theresa May	cameron*
2017	Theresa May	jeremy*
2018	Theresa May	jeremy*
2019	Boris Johnson	jeremy*

Table 5.4: The table shows the names/tokens closest to the vector for *David Cameron* and `cameron` respectively, in 2010. For the token based model, the token in parenthesis is the closest name, shown for a fairer comparison. The asterisk indicates incorrect associations. The token based model is struggling with *Brown*, *May* and *Johnson*. In 2008, it found `gordon`, but not the surname `brown`. While *May* and *Brown* are difficult because they are also common dictionary words, *Johnson* is difficult because it is a common surname.

distance scores for every word to itself throughout the years. The scores can then be sorted, giving a descending list of name, year tuples where the largest changes occurred.

Inspection showed that these detected spikes usually corresponded to long term change events, but also to one-off events in a persons career.

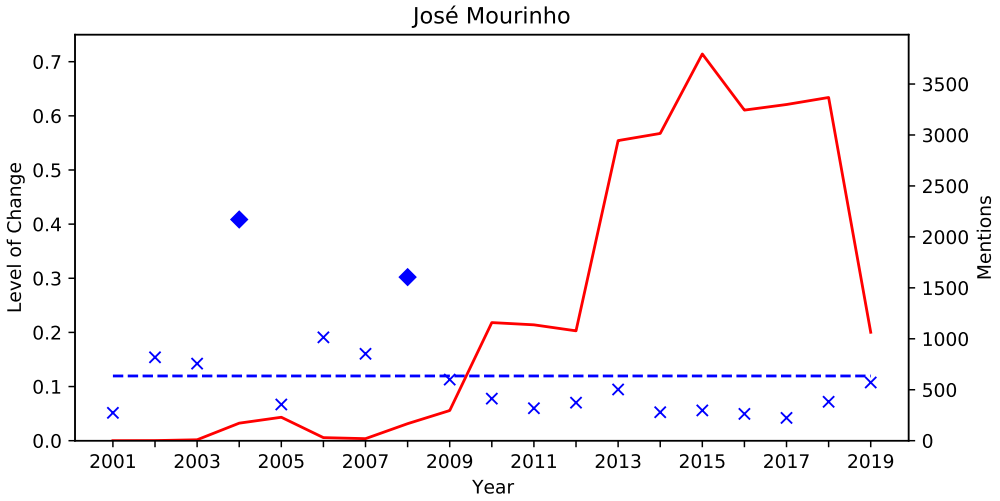

Figure 5.3: The context change and frequency change for José Mourinho. The red line shows the amount of mentions and the blue markers show the cosine distance to the previous year. The dashed line marks the mean change, the diamond markers are more than a standard deviation away from the mean.

Examples of long term change are role changes for politicians or athletes such as the election of *Imran Khan* as prime minister of Pakistan in 2018 or *David Marshall* getting into the Scottish national football team in 2004.

Some spikes corresponded to a significant event, but not one causing a long term change. For example the food writer, chef and cooking show host *Nigella Lawson* was involved in a court case with her partner *Charles Saatchi* in 2013. The aforementioned *Imran Khan* has attended a series of sports event in 2015 which was an unusual context for him. Both of these showed as spikes in the model.

There were also examples of falsely detected change, where two people with the same name from different contexts were treated as the same person. *David Marshall* also has a spike in 2006, which is however due to mentions of a different *David Marshall* in a political context. Similarly there is a musician called *Scott Walker*, but also a US politician. During local elections, reports of his success in the elections show up as spikes in the usually more music-focused reporting about *Scott Walker*.

Overall, spikes are most pronounced in medium to low data settings, where just a few articles can already have a big influence on the context of a person. For example the context changes are for José Mourinho are plotted in Figure 5.3. We can see two big spikes in 2004 and 2008, which is where he first moved from F.C. Porto to Chelsea F.C. and then from Chelsea F.C. to Inter Milan. However, an earlier team change in

2002 to Porto, and subsequent team changes in 2010, 2013 and 2016 are not detected in the model. While in 2002 there is not enough reporting, subsequently José Mourinho might have occurred in too many different context due to him being famous. So detected changes are usually correct, but often many changes are not detected correctly.

5.5.3.1 Comparing to the Baseline

A quick look into the baseline model shows mixed results. The uniqueness of the name is important for finding meaningful results. `Imran` retrieves a trajectory that looks like the one for *Imran Khan*, but the surname does not. Similarly for *Nigella Lawson*, her first name is more distinct than her surname. `Mourinho` and `Saatchi` are easily found and the same spikes are detected, but *Scott Walker* and *David Marshall* are impossible to find, both first and last name are too common.

5.6 Quantitative Analysis

In the qualitative analysis I presented examples of people with common surnames or surnames that are also dictionary words. Here I will quantify these phenomena and attempt to measure the improvements for these persons.

Quantitative analysis is always a difficult task with diachronic models, because there are no gold standard embedding spaces to compare against. Yao et al. [3] use the sections associated with the articles to group words together. They then identify clusters in the embedding space and compare them with the groupings created by the sections. Kutuzov, Velldal, and Øvrelid [2] track countries in the news and use a manually created database of armed conflicts as a target to compare their model against.

I present two experimental setups, one following Yao et al. [3] based on the sections of articles and one inspired by the change event detection by Kutuzov, Velldal, and Øvrelid [2], comparing my model against gold standard change events in the football domain.

5.6.1 Section Analysis

The sections the articles appear in are not used in the model creation process and can therefore be used as a target for evaluation. Based on the sections of the articles, we can calculate the predominant section for each person for each year. We can assume that any two persons that mostly appear in the same section are also closer together in the

vector space of the model. Using spherical k-means clustering we can then cluster the persons in the vector space and compare the clusters to the section groupings. Note that as shown in Section 3.2 the sizes of sections by article count are heavily imbalanced. This translates also to the number of people in the sections. The number of people in a section can differ by two orders of magnitude.

The vectors from all years can be clustered in one big vector space, or clustering can be done for each year and the results averaged over years. The full space emphasises cross-year alignment, whereas the yearly clustering emphasises local separation. I will report results for both as `total` and `yearly`.

To make sure that the person is predominantly associated with a specific section a significant amount of their mentions have to be in a distinct section. Following Yao et al. [3] I used a threshold of 35%, meaning that at least 35% of a persons mentions need to be in a specific section for it to be considered a clear association. To ensure that a reasonable amount of total mentions is available to get high quality embeddings, I included only people that appear at least 500 times in total and each year where they are to be included requires 100 mentions in that year.

The comparison of the `token` and `person` model is difficult because their vocabularies differ and the explicit person names do not appear in the vocabulary of the `token` model. In the token model I used the surname of the person as a marker for them. However, I excluded all people that had a duplicate surname, given that mapping two tokens in the `person` model to a single token in the `token` model would create an inherent error. Note that we only consider people with more than 100 mentions, that means that there might be people with the same surname as someone, but they all have less than 100 mentions, so the only person with more than 100 mentions is considered to have a unique surname.

To evaluate the severity of ambiguous surnames I created two sets:

- `non-ambiguous only`: For every mapped token, I ensured that the mapped token *also* appeared at least 35% of the time in the same section as the full name. This set contains 15,622 vectors.

- `ambiguous names included`: I simply included all the names, regardless of the distribution of the associated token. This set contains 20,084 vectors.

As an example, in 2019 the dominating section for *Theresa May* was `politics` with 63% prevalence. The word `may` appeared mostly in `politics` too, but only with 8% prevalence. *Theresa May* is included in the second set, but not in the first.

	yearly		total	
Clusters	`token` model	`person` model	`token` model	`person` model
non-ambiguous names only				
10	0.6425	**0.6484**	0.6575	**0.6780**
15	0.6565	**0.6731**	0.6602	**0.6671**
20	0.6506	**0.6665**	0.6367	**0.6632**
25	0.6374	**0.6575**	0.6544	**0.6556**
ambiguous names included				
10	0.6472	**0.6478**	**0.6641**	0.6261
15	0.6559	**0.6698**	0.6588	**0.6748**
20	0.6501	**0.6645**	0.6537	**0.6785**
25	0.6400	**0.6586**	0.6523	**0.6630**

Table 5.5: The table shows normalised mutual information (NMI) scores for various experimental settings. The "yearly" column contains clustering results done by year and averaged subsequently. The "total" column contains clustering results of all vectors across all years in one space. Two data sets were evaluated, non-ambiguous names only as well as ambiguous names included. The "total" scores are averaged across three different clusterings, and the "yearly scores" are averaged across three different clusterings per year, yielding 60 scores in total. This was done to smooth out effects of the randomness of the clustering process.

For both the `yearly` and `total` setting, for every clustering I created three clusterings and averaged the results to smooth out effects of random initialisation.

Table 5.5 shows the results of the experiments. Across both data sets and both experimental setups, for various cluster sizes, the `person` model outperforms the `token` model expect in the 10 cluster setup with the ambiguous names included. The margins are small, but scores were averaged over multiple clusterings, especially for the yearly evaluation, making the gains robust.

It is interesting to see that the gains in the set containing the ambiguous names are not larger, as it would be expected. A potential reason could be that while, for example, `may` appears frequently in many sections, the generic contexts "cancel out", leaving the political context given by *Theresa May* as the dominant context for clustering, even though the immediate neighbourhood of the word does not contain words indicating that.

5.6.2 Football

The domain of football lends itself well to an analysis of context changes. There is a relatively large amount of articles on football, the football section is the third biggest section after world news and sports with about 8,000 articles every year. In contrast to politics where there are a number of different topics that can be "in the news", such as immigration, health, the economy or recently Brexit, causing politicians to seemingly change context frequently, people reported about in the football section are usually in the context of their role (player, coach) and team (Manchester United, Chelsea F.C.) and reporting is about matches and team and role changes, with a stable vocabulary of teams and roles. I inspected the phenomenon of players becoming coaches using gold standard data from Wikidata[1].

5.6.2.1 Target Data Retrieval

Wikidata is an open access, community maintained knowledge graph containing over 80 million nodes. To retrieve the relevant persons I first selected people by name and then filtered the list based on specific properties to eliminate duplicates. The base list of names was retrieved from my data, containing any person appearing at least 20 times in the football section. After retrieving the nodes, Certain names were duplicated. To find the football players I looked for the presence of the property "member of sports team" (P54) or "coach of sports team" (P6078) and the value of the property "sport" (P641) had to be "association football" (Q2736). To disambiguate further I prioritised any person that had the property "position played" (P413) or "participant in" (P1344). I reasoned that more complete information was more likely to be available for the person that was reported on, as well as an athlete that already participated in a competition is more likely to be the one that has been reported on in the news.

5.6.2.2 Players Becoming Coaches

I found 39 players that became coaches in the years between 2001 and 2019 that also were present in Wikidata. The full list can be found in the appendix in Table A.1. Based on the assumption that their change from being a player to becoming a coach was their biggest change in their context, I looked for the biggest change in the embedding space throughout all years, in the `person` model. For 8 out of 39 people, their biggest change spike coincided with the year in which they were first a trainer. This gives a 21%

[1] `https://www.wikidata.org/`

accuracy, 4 times better than random. For 5 additional people, a spike occurred that was larger than 1 standard deviation from the mean change, meaning that that change event is not the biggest, but still a significant change in the model.

5.7 Discussion

In the previous sections I looked at the persons that appear in the data and quantified the problem of duplicate names. I showed examples of the representation of people with an explicit name based embedding as well as a token based baseline and analysed the same issue quantitatively using the article sections and clustering of embeddings. To assess the effectiveness of change modelling of such a model I looked at examples of individual changes in the model, as well as looking at a fixed role and observing the changes in people taking this role on the example of the prime minister of the UK. I took a quantitative look at role changes on the example of football players becoming coaches.

Names and duplicates: The analysis of the surnames in the data across the years showed that more than a third of people do not have a unique surname; for all these people their surname does not uniquely identify them and thus any embedding of their surname will not be a good representation of the context of that person, but instead a mix of all the people that share that surname. Especially less mentioned people are overshadowed by other people with the same surname that are mentioned more.

The examples of *Taylor Swift* and the surname *Johnson* support this, showing much more meaningful embeddings for names over tokens. For people that *did* have a unique surname, the embeddings were improved nonetheless, which I showed quantitatively. Improvements were only minor, but consistent across 15 out of 16 different experiments.

Change tracking: For the analysis of the change tracking in the model, the case of the prime minister of the UK showed that the explicit person embeddings improved the association markedly. A systematic look at change detection in role changes for football players showed performance four times better than the random baseline, but there is still a lot of room for improvement.

Change detection for arbitrary people is difficult due to selective reporting in the news. I showed that detected context changes are usually correlated to real world events, but not all events show up in the model. A larger or more specific data set could help with this, but still many events are not reported on. If there is no report of

something in the data, it cannot show up in the diachronic embedding model.

Another effect is that for role changes, the reporting focuses more on the new person overtaking the role than on the old person being replaced. When a prime minister changes, there is a much more noticeable context change for the person getting into office than the person getting out of office.

For athletes, team changes or role changes are also less easy to detect than complete out of context reporting, about for example sexual misconduct or drunk driving. Not only is there a bias in the news to report these things more, but also the words are so different that the context change is much more pronounced in the embedding space than the change from player to coach.

Mention detection: For the detection of names, the proposed approach is limited in its effectiveness for certain grammatical constructs. For example in the expression "The Milibands, Ed and David" there are two people mentioned, *Ed Miliband* and *David Miliband*, however they are not identified with the current method. Then, different names for the same person are not recognized. Examples are *Tony* or *Anthony Blair* and *Kenneth* or *Ken Clarke*. There is also a *José Mourinho* and a *Jose Mourinho*. The model did improve detection for names such as *Brown* or *May* and also for popular names such as *Johnson*, but for a full name that multiple persons have, mistakes are still inevitable. An example is *Scott Walker*, a US politician, but also an American-born British singer-songwriter.

Change classification: Shoemark et al. [8] categorise different types of changes a word can undergo, also distinguishing between actual change of semantics and ephemeral change. It is not straightforward to apply these same categories to people. The changes we observe in the context of a name such as *Barack Obama* are not changes in the semantics of the name; the name always refers to the same person. They are instead changes to the context of the person the name refers to. And changes might be long term context changes due to the person changing their job or similar, but they might also be short term event-based context changes. Nevertheless, for people, these are still interesting, while for words maybe less so.

Chapter 6

Conclusion

Overall, the data analysis and experiments showed that the surname of a person is in many cases not sufficient to identify them in text. I quantified the problem of duplicate surnames and also showed that it is not uncommon for people to have difficult to identify names such as *Swift* or *May*. Using named entity recognition and heuristic name linking I created a model that significantly improved embeddings for persons in the data, even for people that had a unique surname.

For mention detection, besides improvements in named entity recognition a next step would be to add co-reference detection to the model, getting even more contexts for the people that were found. A full blown literary character detection system can be deployed, previous work showed that such systems can significantly improve the number of references that are detected, compared to an NER baseline [17]

The proposed techniques improved the detection of the prime minister of the UK in the embedding model over the token baseline, but the detection of fine grained context changes remains difficult for individuals less reported on. If the model shows a context change, this is usually relatable to a real world event. However, not all real world context changes are reported on, also due to the way the news covers events and reporters select what to report on; not everything is reported on.

The new data set provides a valuable new perspective, complementing previous work with a focus on American news. For example, instead of inspecting the change of the US president I analysed the change of the UK prime minister. The data is easily accessible and extended data sets including more years of data will be easy to create in the future.

Future work should focus on the improvement of named entity recognition, incorporating co-reference resolution into the model and finding and sourcing high quality gold standard context change and event corpora to evaluate models against.

Bibliography

[1] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf (visited on 04/14/2020).

[2] Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. “Tracing Armed Conflicts with Diachronic Word Embedding Models”. In: *Proceedings of the Events and Stories in the News Workshop*. Proceedings of the Events and Stories in the News Workshop. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 31–36. DOI: 10.18653/v1/W17-2705. URL: http://aclweb.org/anthology/W17-2705 (visited on 04/07/2020).

[3] Zijun Yao et al. “Dynamic Word Embeddings for Evolving Semantic Discovery”. In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM ’18* (2018), pp. 673–681. DOI: 10.1145/3159652.3159703. arXiv: 1703.00607. URL: http://arxiv.org/abs/1703.00607 (visited on 04/09/2020).

[4] John R Firth. “A Synopsis of Linguistic Theory, 1930-1955”. In: *Studies in linguistic analysis* (1957).

[5] Omer Levy and Yoav Goldberg. “Neural Word Embedding as Implicit Matrix Factorization”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2177–2185. URL: http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.pdf (visited on 07/21/2020).

[6] Vivek Kulkarni et al. “Statistically Significant Detection of Linguistic Change”. In: (Nov. 12, 2014). arXiv: 1411.3315 [cs]. URL: http://arxiv.org/abs/1411.3315 (visited on 01/22/2020).

[7] Alex Rosenfeld and Katrin Erk. “Deep Neural Models of Semantic Shift”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 474–484. DOI: 10.18653/v1/N18-1044. URL: http://aclweb.org/anthology/N18-1044 (visited on 03/05/2020).

[8] Philippa Shoemark et al. “Room to Glo: A Systematic Comparison of Semantic Change Detection Approaches with Word Embeddings”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019, pp. 66–76. DOI: 10.18653/v1/D19-1007. URL: https://www.aclweb.org/anthology/D19-1007 (visited on 01/30/2020).

[9] J.-B. Michel et al. “Quantitative Analysis of Culture Using Millions of Digitized Books”. In: *Science* 331.6014 (Jan. 14, 2011), pp. 176–182. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1199644. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1199644 (visited on 03/05/2020).

[10] Nikhil Garg et al. “Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes”. In: *Proceedings of the National Academy of Sciences* 115.16 (Apr. 17, 2018), E3635–E3644. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1720347115. pmid: 29615513. URL: https://www.pnas.org/content/115/16/E3635 (visited on 03/05/2020).

[11] Terrence Szymanski. “Temporal Word Analogies: Identifying Lexical Replacement with Diachronic Word Embeddings”. In: *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). ACL 2017. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 448–453. DOI: 10.18653/v1/P17-2071. URL: https://www.aclweb.org/anthology/P17-2071 (visited on 03/09/2020).

[12] Evan Sandhaus. "New York Times Corpus: Corpus Overview". In: *LDC catalogue entry LDC2008T19* (2008). URL: https://catalog.ldc.upenn.edu/LDC2008T19.

[13] Robert Parker et al. *English Gigaword Fifth Edition LDC2011T07*. 2011.

[14] Yating Zhang et al. "Omnia Mutantur, Nihil Interit: Connecting Past with Present by Finding Corresponding Terms across Time". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. ACL-IJCNLP 2015. Beijing, China: Association for Computational Linguistics, July 2015, pp. 645–655. DOI: 10.3115/v1/P15-1063. URL: https://www.aclweb.org/anthology/P15-1063 (visited on 04/06/2020).

[15] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. "Pooled Contextualized Embeddings for Named Entity Recognition". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 724–728. DOI: 10.18653/v1/N19-1078. URL: https://www.aclweb.org/anthology/N19-1078 (visited on 04/07/2020).

[16] Emma Strubell et al. "Fast and Accurate Entity Recognition with Iterated Dilated Convolutions". In: (July 22, 2017). arXiv: 1702.02098 [cs]. URL: http://arxiv.org/abs/1702.02098 (visited on 08/12/2020).

[17] Hardik Vala et al. "Mr. Bennet, His Coachman, and the Archbishop Walk into a Bar but Only One of Them Gets Recognized: On The Difficulty of Detecting Characters in Literary Texts". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 769–774. DOI: 10.18653/v1/D15-1088. URL: http://aclweb.org/anthology/D15-1088 (visited on 04/10/2020).

[18] Kevin Anderson. “Guardian Launches Open Platform Tool to Make Online Content Available Free”. In: *The Guardian* (Mar. 10, 2009). URL: https://www.theguardian.com/media/2009/mar/10/guardian-open-platform.

[19] Junyi Jessy Li, Kapil Thadani, and Amanda Stent. “The Role of Discourse Units in Near-Extractive Summarization”. In: *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. Los Angeles: Association for Computational Linguistics, 2016, pp. 137–147. DOI: 10.18653/v1/W16-3617. URL: http://aclweb.org/anthology/W16-3617 (visited on 03/11/2020).

[20] Nuno Ricardo Pinheiro da Silva Guimarães and Álvaro Pedro de Barros Borges Reis Figueira. “Building a Semi-Supervised Dataset to Train Journalistic Relevance Detection Models”. In: *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*. 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech). Nov. 2017, pp. 1271–1277. DOI: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.204.

[21] Pradeep K. Murukannaiah et al. “Learning a Privacy Incidents Database”. In: *Proceedings of the Hot Topics in Science of Security: Symposium and Boot-camp*. HoTSoS. Hanover, MD, USA: Association for Computing Machinery, Apr. 4, 2017, pp. 35–44. ISBN: 978-1-4503-5274-1. DOI: 10.1145/3055305.3055309. URL: https://doi.org/10.1145/3055305.3055309 (visited on 03/04/2020).

Appendix A

First appendix

Name	Year	Name	Year
Walter Mazzarri	2001	Jürgen Klopp	2015
Didier Deschamps	2001	David Wagner	2015
Roberto Mancini	2001	Zinedine Zidane	2016
Massimiliano Allegri	2003	Patrick Vieira	2016
Ian Rush	2004	Unai Emery	2016
Henning Berg	2005	Olof Mellberg	2016
Paul Gascoigne	2005	Harry Kewell	2017
Gareth Southgate	2006	Thierry Henry	2018
Antonio Conte	2006	Marco Silva	2018
Diego Simeone	2006	Joey Barton	2018
Thomas Tuchel	2007	Sol Campbell	2018
Pep Guardiola	2007	Garry Monk	2018
Paul Le Guen	2007	Jonathan Woodgate	2019
Luis Enrique	2008	Jürgen Klinsmann	2019
Jaap Stam	2009	Scott Parker	2019
Vincenzo Montella	2009	Duncan Ferguson	2019
Mauricio Pochettino	2009	Dick Advocaat	2019
Dietmar Hamann	2011	Mikel Arteta	2019
Laurent Blanc	2013	Pepe Mel	2019
Paul Scholes	2014		

Table A.1: Players that became coaches between 2001 and 2019