Australasian Language Technology Association Workshop 2015

Proceedings of the Workshop

Editors:
Ben Hachey
Kellie Webster

8–9 December 2015
Western Sydney University
Parramatta, Australia

Australalasian Language Technology Association Workshop 2015
(ALTA 2015)

http://www.alta.asn.au/events/alta2015

Online Proceedings:
http://www.alta.asn.au/events/alta2015/proceedings/

Silver Sponsors:

DATA 61 CSIRO

CMCRC ASSURING MARKET QUALITY

Google

Bronze Sponsors:

HUGO

THE UNIVERSITY OF SYDNEY

Volume 13, 2015 ISSN: 1834-7037

ALTA 2015 Workshop Committees

Workshop Co-Chairs

- Ben Hachey (The University of Sydney — Abbrevi8 Pty Ltd)
- Kellie Webster (The University of Sydney)

Workshop Local Organiser

- Dominique Estival (Western Sydney University)
- Caroline Jones (Western Sydney University)

Programme Committee

- Timothy Baldwin (University of Melbourne)
- Julian Brook (University of Toronto)
- Trevor Cohn (University of Melbourne)
- Dominique Estival (University of Western Sydney)
- Gholamreza Haffari (Monash University)
- Nitin Indurkhya (University of New South Wales)
- Sarvnaz Karimi (CSIRO)
- Shervin Malmasi (Macquarie University)
- Meladel Mistica (Intel)
- Diego Mollá (Macquarie University)
- Anthony Nguyen (Australian e-Health Research Centre)
- Joel Nothman (University of Melbourne)
- Cécile Paris (CSIRO – ICT Centre)
- Glen Pink (The University of Sydney)
- Will Radford (Xerox Research Centre Europe)
- Rolf Schwitter (Macquarie University)
- Karin Verspoor (University of Melbourne)
- Ingrid Zukerman (Monash University)

Preface

This volume contains the papers accepted for presentation at the Australasian Language Technology Association Workshop (ALTA) 2015, held at Western Sydney University in Parramatta, Australia on 8–9 December 2015.

The goals of the workshop are to:

- bring together the Language Technology (LT) community in the Australasian region and encourage interactions and collaboration;
- foster interaction between academic and industrial researchers, to encourage dissemination of research results;
- provide a forum for students and young researchers to present their research;
- facilitate the discussion of new and ongoing research and projects;
- increase visibility of LT research in Australasia and overseas and encourage interactions with the wider international LT community.

This year's ALTA Workshop presents 16 peer-reviewed papers, including 12 long papers and 4 short papers. We received a total of 20 submissions for long and short papers. Each paper was reviewed by three members of the program committee. Great care was taken to avoid all conflicts of interest.

ALTA 2015 introduces an experimental presentations track. This aims to encourage broader participation and facilitate local socialisation of international results, including work in progress and work submitted or published elsewhere. Presentations were lightly reviewed by the ALTA executive committee to ensure relevance, with 4 of 5 submissions included in the programme.

ALTA 2015 continues the tradition of including a shared task, this year addressing the identification of French cognates in English text. Participation is summarised in an overview paper by organisers Laurianne Sitbon, Diego Mollá and Haoxing Wang. Participants were invited to submit a system description paper, which are included in this volume without review.

We would like to thank, in no particular order: all of the authors who submitted papers; the programme committee for their valuable time and effort; the local organisers Dominique Estival and Caroline Jones for handling physical logistics and coordination with the Confluence 2015 programme; our keynote speaker Mark Johnson for agreeing to share his perspective on the state of the field; and the panelists Tim Baldwin [Moderator], Grace Chung, David Hawking, Maria Milosavljevic and Doug Oard for agreeing to share their experience and insights. We would like to acknowledge the constant support and advice of the ALTA Executive Committee, and the valuable guidance of previous co-chairs.

Finally, we gratefully recognise our sponsors: Data61/CSIRO, Capital Markets CRC, Google, Hugo/Abbrevi8 and The University of Sydney. Most importantly, their generous support enabled us to offer travel subsidies to all students presenting at ALTA. Their support also funded the conference dinner and student paper awards.

Ben Hachey and Kellie Webster
ALTA Workshop Co-Chairs

ALTA 2015 Programme

* denotes sessions shared with ADCS.

Tuesday, 8 December 2015

Session 1 (Parramatta City Campus, Level 6, Room 9-10)*	
09:15–09:30	Opening remarks
09:30–10:30	Keynote: Doug Oard *Information Abolition*
10:30–11:00	Morning tea
Session 2 (Parramatta City Campus, Level 6, Room 9-10)	
11:00–11:20	Presentation: Trevor Cohn *Unlimited order Language Modeling with Compressed Suffix Trees*
11:20–11:40	Long paper: Caroline Mckinnon, Ibtehal Baazeem and Daniel Angus *How few is too few? Determining the minimum acceptable number of LSA dimensions to visualise text cohesion with Lex*
11:40–12:00	Long paper: Ping Tan, Karin Verspoor and Tim Miller *Structural Alignment as the Basis to Improve Significant Change Detection in Versioned Sentences*
12:00–12:20	Presentation: Kellie Webster and James Curran *Challenges in Resolving Nominal Reference*
12:20–1:20	Lunch
Session 3 (Parramatta City Campus, Level 6, Room 9-10)*	
1:30–2:00	ADCS paper: Viet Phung and Lance De Vine *A study on the use of word embeddings and PageRank for Vietnamese text summarization*
2:00–2:20	Long paper: Mahmood Yousefi Azar, Kairit Sirts, Len Hamey and Diego Mollá *Query-Based Single Document Summarization Using an Ensemble Noisy Auto-Encoder*
2:20–2:40	Long paper: Lan Du, Anish Kumar, Massimiliano Ciaramita and Mark Johnson *Using Entity Information from a Knowledge Base to Improve Relation Extraction*
2:40–2:50	Flash presentation: Hanna Suominen *Preview of CLEF eHealth 2016*
2:50–3:00	Short break
3:00–4:00	Panel: Tim Baldwin [Moderator], Grace Chung, David Hawking, Maria Milosavljevic and Doug Oard *NLP & IR in the Wild*
4:00–4:30	Afternoon tea
Session 4 (Parramatta City Campus, Level 6, Room 9-10)	
4:30–4:50	Long paper: Julio Cesar Salinas Alvarado, Karin Verspoor and Timothy Baldwin *Domain Adaption of Named Entity Recognition to Support Credit Risk Assessment*
4:50–5:10	Presentation: Ben Hachey, Anaïs Cadilhac and Andrew Chisholm *Entity Linking and Summarisation in a News-driven Personal Assistant App*
5:10–5:30	Long paper: Shungwan Kim and Steve Cassidy *Finding Names in Trove: Named Entity Recognition for Australian Historical Newspapers*
6:00	Conference dinner @ Collector Hotel

Wednesday, 9 December 2015

Session 5 (Parramatta South Campus, Rydalmere, Building EA, Room 2.14)	
9:30–9:50	Long paper: Jennifer Biggs *Comparison of Visual and Logical Character Segmentation in Tesseract OCR Language Data for Indic Writing Scripts*
9:50–10:10	Long paper: Daniel Frost and Shunichi Ishihara *Likelihood Ratio-based Forensic Voice Comparison on L2 speakers: A Case of Hong Kong native male production of English vowels*
10:10–10:30	Long paper: Kairit Sirts and Mark Johnson *Do POS Tags Help to Learn Better Morphological Segmentations?*
10:30–11:00	Morning tea
Session 6 (Parramatta South Campus, Rydalmere, Building EA, Room 2.14)	
11:00–11:20	Business Meeting
11:20–11:30	Awards
11:30–11:45	Shared task: Laurianne Sitbon, Diego Mollá and Haoxing Wang *Overview of the 2015 ALTA Shared Task: Identifying French Cognates in English Text*
11:45–12:00	Shared task: Qiongkai Xu, Albert Chen and Chang Li *Detecting English-French Cognates Using Orthographic Edit Distance*
12:00–12:10	Short paper: Fiona Martin and Mark Johnson *More Efficient Topic Modelling Through a Noun Only Approach*
12:10–12:20	Short paper: Dat Quoc Nguyen, Kairit Sirts and Mark Johnson *Improving Topic Coherence with Latent Feature Word Representations in MAP Estimation for Topic Modeling*
12:20–12:30	Short paper: Joel Nothman, Atif Ahmad, Christoph Breidbach, David Malet and Tim Baldwin *Understanding engagement with insurgents through retweet rhetoric*
12:30–12:40	Short paper: Sam Shang Chun Wei and Ben Hachey *A comparison and analysis of models for event trigger detection*
Session 7 (Parramatta South Campus, Rydalmere, Building EE, Foyer)	
12:40–1:30	Lunch
1:30–2:30	Poster session
Session 8 (Parramatta South Campus, Rydalmere, Building EA, Room G.18)*	
2:30–3:30	Keynote: Mark Johnson *Computational Linguistics: The previous and the next five decades*
3:30–4:00	Afternoon tea
Session 8 (Parramatta South Campus, Rydalmere, Building EA, Room 2.14)	
4:00–4:20	Long paper: Hamed Hassanzadeh, Diego Mollá, Tudor Groza, Anthony Nguyen and Jane Hunter *Similarity Metrics for Clustering PubMed Abstracts for Evidence Based Medicine*
4:20–4:40	Long paper: Lance De Vine, Mahnoosh Kholghi, Guido Zuccon, Laurianne Sitbon and Anthony Nguyen *Analysis of Word Embeddings and Sequence Features for Clinical Information Extraction*
4:40–5:00	Long paper: Shervin Malmasi and Hamed Hassanzadeh *Clinical Information Extraction Using Word Representations*
5:00–5:20	Presentation: Andrew MacKinlay, Antonio Jimeno Yepes and Bo Han *A System for Public Health Surveillance using Social Media*
5:20–5:30	ALTA closing

Contents

Long papers **1**

Short papers **110**

ALTA Shared Task papers **133**

Long papers

Query-Based Single Document Summarization Using an Ensemble Noisy Auto-Encoder

Mahmood Yousefi Azar, Kairit Sirts, Diego Mollá Aliod and **Len Hamey**
Department of Computing
Macquarie University, Australia
mahmood.yousefiazar@students.mq.edu.au,
{kairit.sirts, diego.molla-aliod, len.hamey}@mq.edu.au

Abstract

In this paper we use a deep auto-encoder for extractive query-based summarization. We experiment with different input representations in order to overcome the problems stemming from sparse inputs characteristic to linguistic data. In particular, we propose constructing a local vocabulary for each document and adding a small random noise to the input. Also, we propose using inputs with added noise in an Ensemble Noisy Auto-Encoder (ENAE) that combines the top ranked sentences from multiple runs on the same input with different added noise. We test our model on a publicly available email dataset that is specifically designed for text summarization. We show that although an auto-encoder can be a quite effective summarizer, adding noise to the input and running a noisy ensemble can make improvements.

1 Introduction

Recently, deep neural networks have gained popularity in a wide variety of applications, in particular, they have been successfully applied to various natural language processing (NLP) tasks (Collobert et al., 2011; Srivastava and Salakhutdinov, 2012). In this paper we apply a deep neural network to query-based extractive summarization task. Our model uses a deep auto-encoder (AE) (Hinton and Salakhutdinov, 2006) to learn the latent representations for both the query and the sentences in the document and then uses a ranking function to choose certain number of sentences to compose the summary.

Typically, automatic text summarization systems use sparse input representations such as *tf-idf*. However, sparse inputs can be problematic in neural network training and they may make the training slow. We propose two techniques for reducing sparsity. First, we compose for each document a local vocabulary which is then used to construct the input representations for sentences in that document. Second, we add small random noise to the inputs. This technique is similar to the denoising auto-encoders (Vincent et al., 2008). However, the denoising AE adds noise to the inputs only during training, while during test time we also add noise to input.

An additional advantage of adding noise during testing is that we can use the same input with different added noise in an ensemble. Typically, an ensemble learner needs to learn several different models. However, the Ensemble Noisy Auto-Encoder (ENAE) proposed in this paper only needs to train one model and the ensemble is created from applying the model to the same input several times, each time with different added noise.

Text summarization can play an important role in different application domains. For instance, when performing a search in the mailbox according to a keyword, the user could be shown short summaries of the relevant emails. This is especially attractive when using a smart phone with a small screen. We also evaluate our model on a publicly available email dataset (Loza et al., 2014). In addition to summaries, this corpus has also been annotated with keyword phrases. In our experiments we use both the email subjects and annotated keywords as queries.

The contributions of the paper are the following:

1. We introduce an unsupervised approach for extractive summarization using AEs. Although AEs have been previously applied to summarization task as a word filter (Liu et al., 2012), to the best of our knowledge we are the first to use the representations learned by the AE directly for sentence ranking.
2. We add small Gaussian noise to the sparse input representations both during training and

Mahmood Yousefi Azar, Kairit Sirts, Len Hamey and Diego Mollá Aliod. 2015. Query-Based Single Document Summarization Using an Ensemble Noisy Auto-Encoder . In *Proceedings of Australasian Language Technology Association Workshop*, pages 2−10.

testing. To the best of our knowledge, noising the inputs during test time is novel in the application of AEs.

3. We introduce the Ensemble Noisy Auto-Encoder (ENAE) in which the model is trained once and used multiple times on the same input, each time with different added noise.

Our experiments show that although a deep AE can be a quite effective summarizer, adding stochastic noise to the input and running an ensemble on the same input with different added noise can make improvements.

We start by giving the background in section 2. The method is explained in section 3. Section 4 describes the input representations. The Ensemble Noisy Auto-Encoder is introduced in section 5. The experimental setup is detailed in section 6. Section 7 discusses the results and the last section 8 concludes the paper.

2 Background

Automatic summarization can be categorized into two distinct classes: abstractive and extractive. An abstractive summarizer re-generates the extracted content (Radev and McKeown, 1998; Harabagiu and Lacatusu, 2002; Liu et al., 2015). Extractive summarizer, on the other hand, chooses sentences from the original text to be included in the summary using a suitable ranking function (Luhn, 1958; Denil et al., 2014b). Extractive summarization has been more popular due to its relative simplicity compared to the abstractive summarization and this is also the approach taken in this paper.

Both extractive and abstractive summarizers can be designed to perform query-based summarization. A query-based summarizer aims to retrieve and summarize a document or a set of documents satisfying a request for information expressed by a user's query (Daumé III and Marcu, 2006; Tang et al., 2009; Zhong et al., 2015), which greatly facilitates obtaining the required information from large volumes of structured and unstructured data. Indeed, this is the task that the most popular search engines (e.g. Google) are performing when they present the search results, including snippets of text that are related to the query.

There has been some previous work on using deep neural networks for automatic text summarization. The most similar to our work is the model by Zhong et al. (2015) that also uses a deep AE for extractive summarization. However, they use the learned representations for filtering out relevant words for each document which are then used to construct a ranking function over sentences, while we use the learned representations directly in the ranking function. Denil et al. (2014a) propose a supervised model based on a convolutional neural network to extract relevant sentences from documents. Cao et al. (2015) use a recursive neural network for text summarization. However, also their model is supervised and uses hand-crafted word features as inputs while we use an AE for unsupervised learning.

The method of adding noise to the input proposed in this paper is very similar to the denoising auto-encoders (Vincent et al., 2008). In a denoising AE, the input is corrupted and the network tries to undo the effect of the corruption. The intuition is that this rectification can occur if the network learns to capture the dependencies between the inputs. The algorithm adds small noise to the input but the reconstructed output is still the same as uncorrupted input, while our model attempts to reconstruct the noisy input. While denoising AE only uses noisy inputs only in the training phase, we use the input representations with added noise both during training and also later when we use the trained model as a summarizer.

Previously, also re-sampling based methods have been proposed to solve the problem of sparsity for AE (Genest et al., 2011).

3 The Method Description

An AE (Figure 1) is a feed-forward network that learns to reconstruct the input $\boldsymbol{x}$. It first encodes the input $\boldsymbol{x}$ by using a set of recognition weights into a latent feature representation $C(\boldsymbol{x})$ and then decodes this representation back into an approximate input $\hat{\boldsymbol{x}}$ using a set of generative weights.

While most neural-network-based summarization methods are supervised, using an AE provides an unsupervised learning scheme. The general procedure goes as follows:

1. Train the AE on all sentences and queries in the corpus.
2. Use the trained AE to get the latent representations (a.k.a codes or features) for each query and each sentence in the document;
3. Rank the sentences using their latent representations to choose the query-related sentences to be included into the summary.

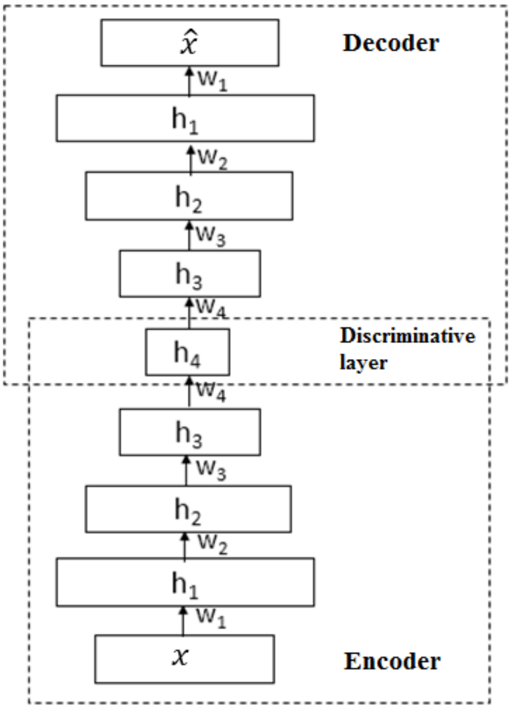


Figure 1: The structure of an AE for dimensionality reduction. x and $\hat{x}$ denote the input and reconstructed inputs respectively. h_i are the hidden layers and w_i are the weights. Features/codes $C(x)$ are in this scheme the output of the hidden layer h_4.

The AE is trained in two phases: *pre-training* and *fine-tuning*. Pre-training performs a greedy layer-wise unsupervised learning. The obtained weights are then used as initial weights in the fine-tuning phase, which will train all the network layers together using back-propagation. The next subsections will describe all procedures in more detail.

3.1 Pre-training Phase

In the pre-training phase, we used restricted Boltzmann machine (RBM) (Hinton et al., 2006). An RBM (Figure 2) is an undirected graphical model with two layers where the units in one layer are observed and in the other layer are hidden. It has symmetric weighted connections between hidden and visible units and no connections between the units of the same layer. In our model, the first layer RBM between the input and the first hidden representation is Gaussian-Bernoulli and the other RBMs are Bernoulli-Bernoulli.

The energy function of a Bernoulli-Bernoulli RBM, i.e. where both observed and hidden units are binary, is bilinear (Hopfield, 1982):

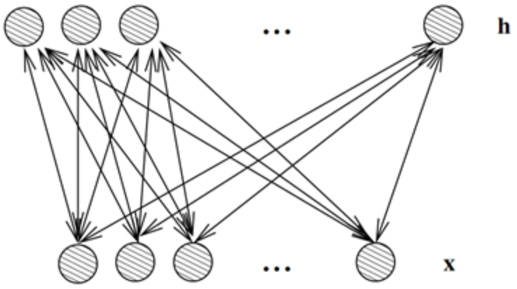


Figure 2: The structure of the restricted Boltzmann machine (RBM) as an undirected graphical model: x denotes the visible nodes and h are the hidden nodes.

$$E(\boldsymbol{x},\boldsymbol{h};\theta) = -\sum_{i\in V} b_i x_i - \sum_{j\in H} a_j h_j - \sum_{i,j} x_i h_j w_{ij}, \tag{1}$$

where V and H are the sets of visible and hidden units respectively, $\boldsymbol{x}$ and $\boldsymbol{h}$ are the input and hidden configurations respectively, w_{ij} is the weight between the visible unit x_i and the hidden unit h_j, and b_i and a_j are their biases. $\theta = \{\mathbf{W},\mathbf{a},\mathbf{b}\}$ denotes the set of all network parameters.

The joint distribution over both the observed and the hidden units has the following equation:

$$p(\boldsymbol{x},\boldsymbol{h};\theta) = \frac{\exp\left(-E(\boldsymbol{x},\boldsymbol{h};\theta)\right)}{Z}, \tag{2}$$

where $Z = \sum_{\boldsymbol{x}',\boldsymbol{h}'} \exp\big(-E(\boldsymbol{x}',\boldsymbol{h}';\theta)\big)$ is the partition function that normalizes the distribution.

The marginal probability of a visible vector is:

$$p(\boldsymbol{x};\theta) = \frac{\sum_{\boldsymbol{h}} \exp\left(-E(\boldsymbol{x},\boldsymbol{h};\theta)\right)}{Z} \tag{3}$$

The conditional probabilities for a Bernoulli-Bernoulli RBM are:

$$p(h_j = 1|\boldsymbol{x};\theta) = \frac{\exp\left(\sum_i w_{ij}x_i + a_j\right)}{1+\exp\left(\sum_i w_{ij}x_i + a_j\right)} = sigm(\sum_i w_{ij}x_i + a_j) \tag{4}$$

$$p(x_i = 1|\boldsymbol{h};\theta) = \frac{\exp\left(\sum_j w_{ij}h_j + b_i\right)}{1+\exp\left(\sum_j w_{ij}h_j + b_i\right)} = sigm(\sum_j w_{ij}h_j + b_i) \tag{5}$$

When the visible units have real values and the hidden units are binary, e.g. the RBM is Gaussian-Bernoulli, the energy function becomes:

$$E(\boldsymbol{x}, \boldsymbol{h}; \theta) = \sum_{i \in V} \frac{(x_i - b_i)^2}{2\sigma_i^2} - \sum_{j \in H} a_j h_j - \sum_{i,j} \frac{x_i}{\sigma_i} h_j w_{ij}, \tag{6}$$

where σ_i is the standard deviation of the ith visible unit. With unit-variance the conditional probabilities are:

$$\begin{aligned} p(h_j = 1 | \boldsymbol{x}; \theta) &= \frac{\exp\left(\sum_i w_{ij} x_i + a_j\right)}{1 + \exp\left(\sum_i w_{ij} x_i + a_j\right)} \\ &= sigm(\sum_i w_{ij} x_i + a_j) \end{aligned} \tag{7}$$

$$\begin{aligned} p(x_i | \boldsymbol{h}; \theta) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x - b_i - \sum_j w_{ij} h_j)^2}{2}\right) \\ &= \mathcal{N}\left(\sum_j w_{ij} h_j + b_i, 1\right) \end{aligned} \tag{8}$$

To estimate the parameters of the network, maximum likelihood estimation (equivalent to minimizing the negative log-likelihood) can be applied. Taking the derivative of the negative log-probability of the inputs with respect to the weights leads to a learning algorithm where the update rule for the weights of a RBM is given by:

$$\Delta w_{ij} = \epsilon(\langle x_i, h_j \rangle_{data} - \langle x_i, h_j \rangle_{model}), \tag{9}$$

where ϵ is the learning rate, angle brackets denote the expectations and $\langle x_i, h_j \rangle_{data}$ is the so-called positive phase contribution and $\langle x_i, h_j \rangle_{model}$ is the so-called negative phase contribution. In particular, the positive phase is trying to decrease the energy of the observation and the negative phase increases the energy defined by the model. We use k-step contrastive divergence (Hinton, 2002) to approximate the expectation defined by the model. We only run one step of the Gibbs sampler, which provides low computational complexity and is enough to get a good approximation.

The RBM blocks can be stacked to form the topology of the desired AE. During pre-training the AE is trained greedily layer-wise using individual RBMs, where the output of one trained RBM is used as input for the next upper layer RBM (Figure 3).

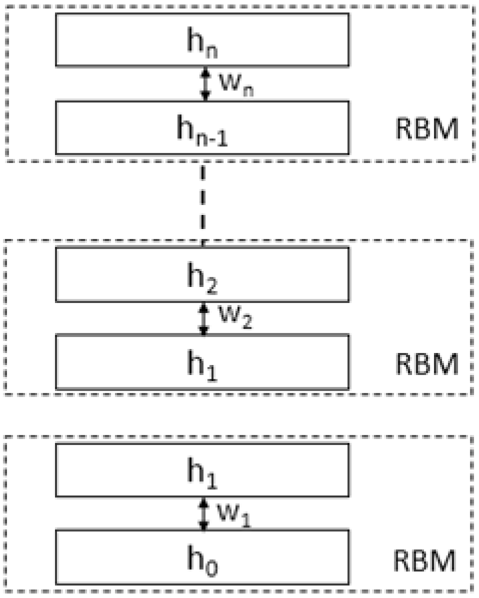


Figure 3: Several generative RBM models stacked on top of each other.

3.2 Fine-tuning Phase

In this phase, the weights obtained from the pre-training are used to initialise the deep AE. For that purpose, the individual RBMs are stacked on top of each other and unrolled, i.e. the recognition and generation weights are tied.

Ngiam et al. (2011) evaluated different types of optimization algorithm included stochastic gradient descent (SGD) and Conjugate gradient (CG). It has been observed that mini-batch CG with line search can simplify and speed up different types of AEs compared to SGD. In this phase, the weights of the entire network are fine-tuned with CG algorithm using back-propagation. The cost function to be minimised is the cross-entropy error between the given and reconstructed inputs.

3.3 Sentence Ranking

Extractive text summarization is also known as sentence ranking. Once the AE model has been trained, it can be used to extract the latent representations for each sentence in each document and for each query. We assume that the AE will place the sentences with similar semantic meaning close to each other in the latent space and thus, we can use those representations to rank the sentences accord-

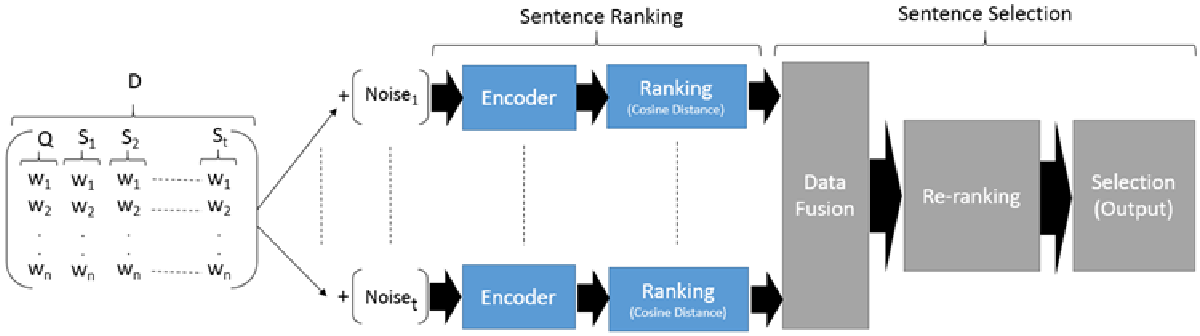


Figure 4: The Ensemble Noisy Auto-Encoder.

ing to their relevance to the query. We use cosine similarity to create the ranking ordering between sentences.

4 Input Representations

The most common input representation used in informations retrieval and text summarization systems is *tf-idf* (Wu et al., 2008), which represents each word in the document using its term frequency *tf* in the document, as well as over all documents (*idf*). In the context of text summarization the *tf-idf* representations are constructed for each sentence. This means that the input vectors are very sparse because each sentence only contains a small number of words.

To address the sparsity, we propose computing the *tf* representations using local vocabularies. We construct the vocabulary for each document separately from the most frequent terms occurring in that document. We use the same number of words in the vocabulary for each document.

This local representation is less sparse compared to the *tf-idf* because the dimensions in the input now correspond to words that all occur in the current document. Due to the local vocabularies the AE input dimensions now correspond to different words in different documents. As a consequence, the AE positions the sentences of different documents into different semantic subspaces. However, this behaviour causes no adverse effects because our system extracts each summary based on a single document only.

In order to reduce the sparsity even more, we add small Gaussian noise to the input. The idea is that when the noise is small, the information in the noisy inputs is essentially the same as in the input vectors without noise.

5 Ensemble Noisy Auto-Encoder

After ranking, a number of sentences must be selected to be included into the summary. A straightforward selection strategy adopted in most extractive summarization systems is just to use the top ranked sentences. However, we propose a more complex selection strategy that exploits the noisy input representations introduced in the previous section. By adding random noise to the input we can repeat the experiment several times using the same input but with different noise. Each of those experiments potentially produces a slightly different ranking, which can be aggregated into an ensemble.

In particular, after running the sentence ranking procedure multiple times, each time with different noise in the input, we use a voting scheme for aggregating the ranks. In this way we obtain the final ranking which is then used for the sentence selection. The voting scheme counts, how many times each sentence appears in all different rankings in the n top positions, where n is a predefined parameter. Currently, we use the simple counting and do not take into account the exact position of the sentence in each of the top rankings. Based on those counts we produce another ranking over only those sentences that appeared in the top rankings of the ensemble runs. Finally, we just select the top sentences according to the final ranking to produce the summary.

A detailed schematic of the full model is presented in Figure 4. The main difference between the proposed approach and the commonly used ensemble methods lies in the number of trained models. Whereas during ensemble learning several different models are trained, our proposed approach only needs to train a single model and the ensemble is created by applying it to a single input repeatedly, each time perturbing it with different noise.

6 Experimental Setup

We perform experiments on a general-purpose summarization and keyword extraction dataset (SKE) (Loza et al., 2014) that has been annotated with both extractive and abstractive summaries, and additionally also with keyword phrases. It consists of 349 emails from which 319 have been selected from the Enron email corpus and 30 emails were provided by the volunteers. The corpus contains both single emails and email threads that all have been manually annotated by two different annotators.

We conduct two different experiments on the SKE corpus. First, we generate summaries based on the subject of each email. As some emails in the corpus have empty subjects we could perform this experiment only on the subset of 289 emails that have non-empty subjects. Secondly, we generate summaries using the annotated keyword phrases as queries. As all emails in the corpus have been annotated with keyword phrases, this experiment was performed on the whole dataset. The annotated extractive summaries contain 5 sentences and thus we also generate 5 sentence summaries.

ROUGE (Lin, 2004) is the fully automatic metric commonly used to evaluate the text summarization results. In particular, ROUGE-2 recall has been shown to correlate most highly with human evaluator judgements (Dang and Owczarzak, 2008). We used 10-fold cross-validation, set the confidence interval to 95% and used the jackknifing procedure for multi-annotation evaluation (Lin, 2004).

Our deep AE implementation is based on G. Hinton's software, which is publicly available.[1] We used mini-batch gradient descent learning in both pre-training and fine-tuning phases. The batch size was 100 data items during pre-training and 1000 data items during fine-tuning phase. During pre-training we trained a 140-40-30-10 network with RBMs and in fine-tuning phase we trained a 140-40-30-10-30-40-140 network as the AE. Here, 140 is the size of the first hidden layer and 10 is the size of the sentence representation layer, which is used in the ranking function.

As a pre-processing step, we stem the documents with the Porter stemmer and remove the stop words.[2]

Model	Subject	Phrases
tf-idf V=1000	0.2312	0.4845
tf-idf V=5%	0.1838	0.4217
tf-idf V=2%	0.1435	0.3166
tf-idf V=60	0.1068	0.2224
AE (*tf-idf* V=2%)	0.3580	0.4795
AE (*tf-idf* V=60)	0.3913	0.4220
L-AE	0.4948	0.5657
L-NAE	0.4664	0.5179
L-ENAE	0.5031	0.5370

Table 1: ROUGE-2 recall for both subject-oriented and key-phrase-oriented summarization. The upper section of the table shows *tf-idf* baselines with various vocabulary sizes. The middle section shows AE with *tf-idf* as input representations. The bottom section shows the AE with input representations constructed using local vocabularies (L-AE), L-AE with noisy inputs (L-NAE) and the Ensemble Noisy AE (L-ENAE).

We use *tf-idf*[3] as the baseline. After preprocessing, the SKE corpus contains 6423 unique terms and we constructed *tf-idf* vectors based on the 1000, 320 (5% of the whole vocabulary), 128 (2% of the whole vocabulary), and 60 most frequently occurring terms. $V = 60$ is the size of the *tf* representation used in our AE model. [4]

We apply the AE model to several different input representations: *tf-idf*, *tf* constructed using local vocabularies as explained in Section 4 (L-AE), *tf* using local vocabularies with added Gaussian noise (L-NAE) and in the noisy ensemble (L-ENAE).

7 Results and Discussion

Table 1 shows the ROUGE-2 scores of the *tf-idf* baselines and the AE model with various input representations. The columns show the scores of the summaries generated using the subjects and keyword phrases as queries respectively.

The main thing to note is that AE performs in most cases much better than the *tf-idf* baseline, especially when using subjects as queries. The only scenario where the *tf-idf* can compete with the AE

[1] `http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html`

[2] Stop word list obtained from `http://xpo6.com/list-of-english-stop-words`

[3] We use the inverse frequency as the *idf* term.

[4] We did not train the AE-s with larger vocabularies because this would have required changing the network structure as during the preliminary experiments we noticed that the network did not improve much when using inputs larger than the first hidden layer.

is with the vocabulary of size 1000 and when using keyword phrases as queries. This is because the manually annotated keyword phrases do in many cases contain the same words as extracted summary sentences, especially because the queries and summaries where annotated by the same annotators. However, when the vocabulary size is the same as used in the AE, *tf-idf* scores are much lower in both experimental settings.

The second thing to notice is that although AE with *tf-idf* input performs better than plain *tf-idf*, it is still quite a bit worse than AE using *tf* representations derived from the local vocabularies. We believe it is so because the deep AE can extract additional information from the *tf-idf* representations, but the AE learning is more effective when using less sparse inputs, provided by the *tf* representations constructed from local vocabularies.

Although we hoped that the reduced sparsity stemming from the added noise will improve the results even more, the experiments show that this is not the case—AE without noise performs in both settings better than the noisy AE. However, when combining the rankings of the noisy ensemble, the results are much better than a single noisy AE and may even improve over the simple AE. This is the case when extracting summaries based on the subject. The subjects of the emails are less informative than the annotated keyword phrases. Perhaps this explains why the ENAE was able to make use of the noisy inputs to gain small improvements over the AE without any noise in this scenario.

There is considerable difference between the results when using the email subjects or keyword phrases as queries with keyword phrases leading to better summaries. This is to be expected because the keyword phrases have been carefully extracted by the annotators. The keyword phrases give the highest positive contribution to the *td-idf* baselines with largest vocabularies, which clearly benefits from the fact that the annotated sentences contain the extracted keyword phrases. The ENAE shows the smallest difference between the subject-based and keyword-based summaries. We believe it is because the ENAE is able to make better use of the whole latent semantic space of the document to extract the relevant sentences to the query, regardless of whether the query contains the exact relevant terms or not.

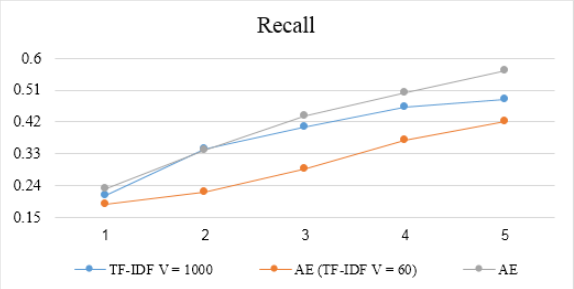


Figure 5: ROUGE-2 recall for summaries containing different number of sentences using the keyword phrases as queries.

Figure 5 illustrates the ROUGE-2 recall of the best baseline and the AE models with both *tf-idf* and *tf* input representations using keyword phrases as queries and varying the length of the generated summaries. In this experiment, each summary was evaluated against the annotated summary of the same length. As is expected, the results improve when the length of the summary increases. While the AE model's results improve almost linearly over the 5 sentences, *tf-idf* gains less from increasing the summary length from 4 to 5 sentences. The scores are almost the same for the *tf-idf* and the AE with *tf* representation with 1 and 2 sentences. Starting from 3 sentences, the AE performs clearly better.

To get a better feel what kind of summaries the ENAE system is generating we present the results of a sample email thread (ECT020). This typical email thread contains 4 emails and 13 lines. The summaries extracted by the ENAE system using both subjects and keyword phrases are given in Figure 6. The annotated summaries consist of sentences [03, 04, 10, 11, 05] and [03, 04, 11, 06, 05] for the first and the second annotator respectively.

Both generated summaries contain the sentences 03, 04 and 06. These were also the sentences chosen by the annotators (03 and 04 by the first annotation and all three of them by the second). The sentence 11 present in the subject-based summary was also chosen by both annotators, while sentence 10 in keyword-based summary was also annotated by the first annotator. The only sentences that were not chosen by the annotators are 08 in the subject-based summary and 12 in the keyword-based summary. Both annotators had also chosen sentence 05, which is not present in the automatically generated summaries. However, this is the sentence that both annotators gave the last priority in their rankings.

In general the order of the sentences generated by the system and chosen by the annotators is the

a) ENAE summary based on subject

03 Diamond-san, As I wrote in the past, Nissho Iwai's LNG related department has been transferred into a new joint venture company between Nissho and Sumitomo Corp. as of October 1, 2001, namely, "LNG Japan Corp.".

08 We are internally discussing when we start our official meeting.

04 In this connection, we would like to conclude another NDA with LNG Japan Corp, as per attached.

06 Also, please advise us how we should treat Nissho's NDA in these circumstances.

11 They need to change the counterparty name due to a joint venture.

b) ENAE summary based on keyword phrases

10 Please approve or make changes to their new NDA.

03 Diamond-san, As I wrote in the past, Nissho Iwai's LNG related department has been transferred into a new joint venture company between Nissho and Sumitomo Corp. as of October 1, 2001, namely, "LNG Japan Corp.".

04 In this connection, we would like to conclude another NDA with LNG Japan Corp, as per attached.

12 I wanted to let you know this was coming in as soon as Mark approves the changes.

06 Also, please advise us how we should treat Nissho's NDA in these circumstances.

Figure 6: Examples of subject-based (left) and keyword-based (right) summaries extrated by the Ensemble Noisy AE.

a) First annotator

LNG Japan Corp. is a new joint venture between Nissho and Sumitomo Corp. Given this situation a new NDA is needed and sent for signature to Daniel Diamond. Daniel forward the NDA to Mark for revision.

b) Second annotator

An Enron employee is informed by an employee of Nissho Iwai that the Nissho Iwai's LNG related department has been transferred into a new joint venture company, namely, 'LNG Japan Corp.'. As a result, there is a need to change the counterparty name in the new NDA. The new change has to be approved and then applied to the new NDA with LNG Japan Corporation

Figure 7: The abstractive summaries created by the annotators for the example email.

same in both example summaries. The only exception is sentence 10, which is ranked as top in the summary generated based on the keyword phrases but chosen as third after the sentences 03 and 04 by the first annotator.

Looking at the annotated abstractive summaries we found that the sentence 12 chosen by the keyword-based summarizer is not a fault extraction. Although neither of the annotators chose this sentence for the extractive summary, the information conveyed in this sentence can be found in both annotated abstractive summaries (Figure 7).

8 Conclusion

In this paper we used a deep auto-encoder (AE) for query-based extractive summarization. We tested our method on a publicly available email dataset and showed that the auto-encoder-based models perform much better than the *tf-idf* baseline. We proposed using local vocabularies to construct input representations and showed that this improves over the commonly used *tf-idf*, even when the latter is used as input to an AE. We proposed adding small stochastic noise to the input representations to reduce sparsity and showed that constructing an ensemble by running the AE on the same input multiple times, each time with different noise, can improve the results over the deterministic AE.

In future, we plan to compare the proposed system with the denoising auto-encoder, as well as experiment with different network structures and vocabulary sizes. Also, we intend to test our Ensemble Noisy Auto-Encoder on various different datasets to explore the accuracy and stability of the method more thoroughly.

References

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Ronan Collobert, Jason Weston, Lon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the tac 2008 update summarization task. In *Proceedings of text analysis conference*, pages 1–16.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of ACL06*, pages 305–312. Association for Computational Linguistics.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2014a. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*.

Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, and Nando de Freitas. 2014b. Modelling, visualising and summarising documents with a single convolutional neural network. *arXiv preprint arXiv:1406.3830*.

Pierre-Etienne Genest, Fabrizio Gotti, and Yoshua Bengio. 2011. Deep learning for automatic summary scoring. In *Proceedings of the Workshop on Automatic Text Summarization*, pages 17–28.

Sanda M Harabagiu and Finley Lacatusu. 2002. Generating single and multi-document summaries with gistexter. In *Document Understanding Conferences*.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

John J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8.

Yan Liu, Sheng-hua Zhong, and Wenjie Li. 2012. Query-oriented multi-document summarization via unsupervised deep learning. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations.

Vanessa Loza, Shibamouli Lahiri, Rada Mihalcea, and Po-Hsiang Lai. 2014. Building a dataset for summarization and keyword extraction from emails. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

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

Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, Quoc V Le, and Andrew Y Ng. 2011. On optimization methods for deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 265–272.

Dragomir R Radev and Kathleen R McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):470–500.

Nitish Srivastava and Ruslan R Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230.

Jie Tang, Limin Yao, and Dewei Chen. 2009. Multi-topic based query-oriented summarization. In *SDM*, volume 9, pages 1147–1158. SIAM.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):13.

Sheng-hua Zhong, Yan Liu, Bin Li, and Jing Long. 2015. Query-oriented unsupervised multi-document summarization via deep learning model. *Expert Systems with Applications*, 42(21):8146–8155.

Comparison of Visual and Logical Character Segmentation in Tesseract OCR Language Data for Indic Writing Scripts

Jennifer Biggs
National Security & Intelligence, Surveillance & Reconnaissance Division
Defence Science and Technology Group
Edinburgh, South Australia
{firstname.lastname@dsto.defence.gov.au}

Abstract

Language data for the Tesseract OCR system currently supports recognition of a number of languages written in Indic writing scripts. An initial study is described to create comparable data for Tesseract training and evaluation based on two approaches to character segmentation of Indic scripts; logical vs. visual. Results indicate further investigation of visual based character segmentation language data for Tesseract may be warranted.

1 Introduction

The Tesseract Optical Character Recognition (OCR) engine originally developed by Hewlett-Packard between 1984 and 1994 was one of the top 3 engines in the 1995 UNLV Accuracy test as “HP Labs OCR” (Rice et al 1995). Between 1995 and 2005 there was little activity in Tesseract, until it was open sourced by HP and UNLV. It was re-released to the open source community in August of 2006 by Google (Vincent, 2006), hosted under Google code and GitHub under the *tesseract-ocr* project.[1] More recent evaluations have found Tesseract to perform well in comparisons with other commercial and open source OCR systems (Dhiman and Singh. 2013; Chattopadhyay et al. 2011; Heliński et al. 2012; Patel et al. 2012; Vijayarani and Sakila. 2015). A wide range of external tools, wrappers and add-on projects are also available including Tesseract user interfaces, online services, training and training data preparation, and additional language data.

Originally developed for recognition of English text, Smith (2007), Smith et al (2009) and Smith (2014) provide overviews of the Tesseract system during the process of development and internationalization. Currently, Tesseract v3.02 release, v3.03 candidate release and v3.04 development versions are available, and the *tesseract-ocr* project supports recognition of over 60 languages.

Languages that use Indic scripts are found throughout South Asia, Southeast Asia, and parts of Central and East Asia. Indic scripts descend from the Brāhmī script of ancient India, and are broadly divided into North and South. With some exceptions, South Indic scripts are very rounded, while North Indic scripts are less rounded. North Indic scripts typically incorporate a horizontal bar grouping letters.

This paper describes an initial study investigating alternate approaches to segmenting characters in preparing language data for Indic writing scripts for Tesseract; logical and a visual segmentation. Algorithmic methods for character segmentation in image processing are outside of the scope of this paper.

2 Background

As discussed in relation to several Indian languages by Govandaraju and Stelur (2009), OCR of Indic scripts presents challenges which are different to those of Latin or Oriental scripts. Recently there has been significantly more progress, particularly in Indian languages (Krishnan et al 2014; Govandaraju and Stelur. 2009; Yadav et al. 2013). Sok and Taing (2014) describe recent research in OCR system development for Khmer, Pujari and Majhi (2015) provide a survey

[1] The *tesseract-ocr* project repository was archived in August 2015. The main repository has moved from https://code.google.com/p/tesseract-ocr/ to https://github.com/tesseract-ocr

Jennifer Biggs. 2015. Comparison of Visual and Logical Character Segmentation in Tesseract OCR Language Data for Indic Writing Scripts . In *Proceedings of Australasian Language Technology Association Workshop*, pages 11−20.

of Odia character recognition, as do Nishad and Bindu (2013) for Malayalam.

Except in cases such as Krishnan et al. (2014), where OCR systems are trained for whole word recognition in several Indian languages, character segmentation must accommodate inherent characteristics such as non-causal (bidirectional) dependencies when encoded in Unicode.[2]

2.1 Indic scripts and Unicode encoding

Indic scripts are a family of abugida writing systems. Abugida, or alphasyllabary, writing systems are partly syllabic, partly alphabetic writing systems in which consonant-vowel sequences may be combined and written as a unit. Two general characteristics of most Indic scripts that are significant for the purposes of this study are that:

- Diacritics and dependent signs might be added above, below, left, right, around, surrounding or within a base consonant.
- Combination of consonants without intervening vowels in ligatures or noted by special marks, known as consonant clusters.

The typical approach for Unicode encoding of Indic scripts is to encode the consonant followed by any vowels or dependent forms in a specified order. Consonant clusters are typically encoded by using a specific letter between two consonants, which might also then include further vowels or dependent signs. Therefore the visual order of graphemes may differ from the logical order of the character encoding. Exceptions to this are Thai, Lao (Unicode v1.0, 1991) and Tai Viet (Unicode v5.2, 2009), which use visual instead of logical order. New Tai Lue has also been changed to a visual encoding model in Unicode v8.0 (2015, Chapter 16). Complex text rendering may also contextually shape characters or create ligatures. Therefore a Unicode character may not have a visual representation within a glyph, or may differ from its visual representation within another glyph.

2.2 Tesseract

As noted by White (2013), Tesseract has no internal representations for diacritic marks. A typical OCR approach for Tesseract is therefore to train for recognition of the combination of characters including diacritic marks. White (2013) also notes that diacritic marks are often a common source of errors due to their small size and distance from the main character, and that training in a combined approach also greatly expands the larger OCR character set. This in turn may also increase the number of similar symbols, as each set of diacritic marks is applied to each consonant.

As described by Smith (2014), lexical resources are utilised by Tesseract during two-pass classification, and de Does and Depuydt (2012) found that word recall was improved for a Dutch historical recognition task by simply substituting the default Dutch Tesseract v3.01 word list for a corpus specific word list. As noted by White (2013), while language data was available from the *tesseract-ocr* project, the associated training files were previously available. However, the Tesseract project now hosts related files from which training data may be created.

Tesseract is flexible and supports a large number of control parameters, which may be specified via a configuration file, by the command line interface, or within a language data file[3]. Although documentation of control parameters by the *tesseract-ocr* project is limited[4], a full list of parameters for v3.02 is available[5]. White (2012) and Ibrahim (2014) describe effects of a limited number of control parameters.

2.2.1 Tesseract and Indic scripts

Training Tesseract has been described for a number of languages and purposes (White, 2013; Mishra et al. 2012; Ibrahim, 2014; Heliński et al. 2012). At the time of writing, we are aware of a number of publically available sources for Tesseract language data supporting Indic scripts in addition to the *tesseract-ocr* project. These include *Parichit*[6], *BanglaOCR*[7] (Hasnat et al. 2009a and 2009b; Omee et al. 2011) with training files released in 2013, *tesseractindic*[8], and *myaocr*[9]. Their Tesseract version and recognition languages are summarised in Table 1. These external projects also provide Tesseract training data in the form of TIFF image and associated coordinate 'box' files. For version 3.04, the *tesseract-ocr* project provides data from which Tesseract can generate training data.

[2] Except in Thai, Lao, Tai Viet, and New Tai Lue

[3] Language data files are in the form <xxx>.traineddata

[4] https://code.google.com/p/tesseract-ocr/wiki/ControlParams

[5] http://www.sk-spell.sk.cx/tesseract-ocr-parameters-in-302-version

[6] https://code.google.com/p/Parichit/

[7] https://code.google.com/p/banglaocr/

[8] https://code.google.com/p/tesseractindic/

[9] https://code.google.com/p/myaocr/

Sets of Tesseract language data for a given language may differ significantly in parameters including coverage of the writing script, fonts, number of training examples, or dictionary data.

Project	v.	Languages
tesseract-ocr	3.04	Assamese, Bengali, Gujarati, Hindi, Marathi, Odia, Punjabi, Tamil, Myanmar, Khmer, Lao, Thai, Sinhala, Malayalam, Kannada, Telugu
	3.02	Bengali, Tamil, Thai
	3.01	Hindi, Thai
myaocr	3.02	Myanmar
Parichit	3.01	Bengali, Gujarati, Hindi, Oriya, Punjabi, Tamil, Malayalam, Kannada, Telugu
tesseractindic	2.04	Hindi, Bengali, Malayalam
BanglaOCR	2	Bengali

Table 1: Available Indic language data for Tesseract

Smith (2014)[10] and Smith et al (2009)[11] provides results for Tesseract for two Indic scripts; Hindi[12] and Thai. Table 2 compares these error rates to those found by Krishnan et al. (2014)[13]. Additionally, the Khmer OCR project reports initial accuracy rates of 50-60% for Khmer OS Battambang font, 26pt (Tan, 2014), and the Khmer OCR project[14] beta website provides a Khmer OCR web service based on the Tesseract OCR system that incorporates user feedback training. Hasnat et al. (2009a; 2009b) report on development of Bengali language data for BanglaOCR, with 70-93% accuracy depending on image type. Omee et al. (2011) report up to 98% accuracy in limited contexts for BanglaOCR. Nayak and Nayak (2014) report on development of Odia language data with 98-100% recognition accuracy for isolated characters.

Language	Ground truth (million)		Error rate (%)	
	char	words	char	word
Hindi *	-	0.39	26.67	42.53
Telugu *	-	0.2	32.95	72.11
Hindi **	2.1	0.41	6.43	28.62
Thai **	0.19	0.01	21.31	80.53
Hindi ***	1.4	0.33	15.41	69.44

Table 2: Tesseract error rates * from Krishnan et al. (2014) ** from Smith (2014) * from Smith et al (2009)**

2.2.2 Visual and logical character segmentation for Tesseract

As noted by White (2013) the approach of the *tesseract-ocr* project is to train Tesseract for recognition of combinations of characters including diacritics. For languages with Indic writing scripts, this approach may also include consonant-vowel combinations and consonant clusters with other dependent signs, and relies on character segmentation to occur in line with Unicode logical ordering segmentation points for a given segment of text. An advantage of this approach is that Unicode standard encoding is output by the OCR system.

An alternate approach in developing a training set for Tesseract is to determine visual segmentation points within the writing script. This approach has been described and implemented in several external language data projects for Tesseract, including *Parichit*, *BanglaOCR*, and *myaocr*. Examples of logical and two possible approaches to visual segmentation for selected consonant groupings are shown in Figure 1. A disadvantage of visual segmentation is that OCR text outputs may require re-ordering processing to output Unicode encoded text.

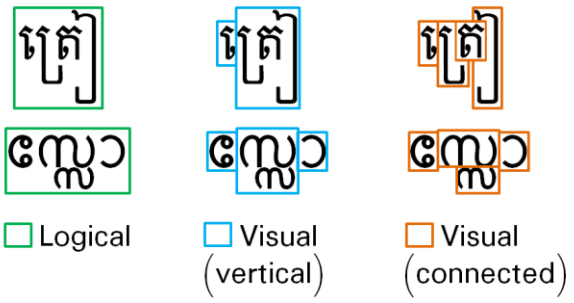


Figure 1: Comparison of logical and two possible visual segmentation approaches for selected characters

[10] Tesseract v3.03 or v3.04

[11] Tesseract v3.00

[12] Hindi and Arabic language data for Tesseract v3.02 used a standard conventional neural network character classifier in a ‘cube’ model. Although, Smith (2014) states that this model achieves ~50% reduction in errors on Hindi when run together with Tesseract’s word recognizer, the training code is unmaintained and unutilised, and will be removed from future *tesseract-ocr* versions.

[13] Tesseract v3.02

[14] The Khmer OCR project led by Mr. Danh Hong begun in 2012 is described by Mr. Ly Sovannra in Tan (2014) and at http://www.khmertype.org

Mishra et al. (2012) describe creating language data for Hindi written in Devanagari script that implemented a visual segmentation approach in which single touching conjunct characters are excluded from the training set. Therefore, Tesseract language data could be created that included only two or more touching conjunct characters, basic characters and isolated half characters. This had the effect of reducing the Tesseract training set[15] and language data size, and increasing recognition accuracy on a test set of 94 characters compared with the *tesseract-ocr* (Google) and *Parichit* language data as shown in Table 3.[16]

Language data	Training set size	Accuracy (%)
tesseract-ocr v3.01	1729	45.2
Parichit	2173	22.3
Mishra et al. (2012)	786	90.9

Table 3: Comparison of training set, language data and accuracy from Mishra et al. (2012)

The implementation also included language-specific image pre-processing to ‘chop’ the *Shirorekha* horizontal bar connecting characters within words. This was intended to increase the likelihood of Tesseract system segmentation occurring at these points. Examples of words including *Shirorekha* are shown in Figure 2.

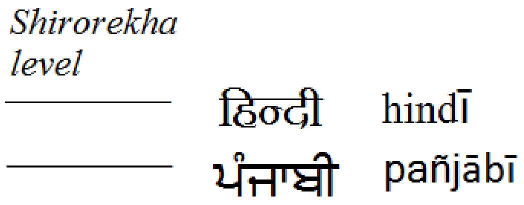


Figure 2: Examples of *Shirorekha* in Devanagari and Gurmukhi scripts

3 Comparison of visual and logical segmentation for Tesseract

An initial study was conducted to determine the potential of implementing a visual segmentation approach, compared to the logical segmentation approach in Tesseract for languages with Indic scripts. Languages written with Indic scripts that do not use the *Shirorekha* horizontal bar were considered. Re-ordering of OCR text outputs for visual segmentation methods is outside the scope of this study. The term glyph is used in this section to describe a symbol that represents an OCR recognition character, whether by logical or visual segmentation.

3.1 Method

This section describes ground truth and evaluation tools used, and the collection and preparation of glyph, Tesseract training, and OCR ground truth data. Three Indic languages were selected to estimate the potential for applying visual segmentation to further languages. Firstly, corpora were collected and analysed to compare glyphs found by each segmentation approach. Secondly, Tesseract recognition and layout accuracy was evaluated based on the coverage of those glyphs in the corpus. The accuracy of *tesseract-ocr* project v3.04 language data is also measured against the same ground truth data for a wider selection of Indic languages.

3.1.1 Glyph data

In order to estimate the number and distribution of glyphs in selected Indic languages, language specific corpora were sought. A web crawler was implemented using the *crawler4j* library[17], which restricted the crawl domain to the seed URL. The *boilerpipe* library [18] was then used to extract textual content from each web page. For each language, a corpus was then collected by using the relevant Wikipedia local language top page as the seed for the crawler.

The *Lucene* library[19] was used to index corpus documents. Language specific processing was implemented supporting grouping of consonant-vowel combinations, consonant clusters and dependent signs into logical order glyphs. Additional processing to separate those groupings in line with the visual segmentation approach was also implemented.

Letters affected by visual segmentation in each language are shown in Table 4. In Khmer, there could theoretically be up to three *coeng* (U+17D2) in a syllable; two before and one after a vowel. Clusters with *coeng* after a vowel were not additionally segmented in this implementation. The number of glyphs according to each segmentation approach was then extracted from the index for each language. Similarly, in Mala-

[15] Defined in Tesseract the *.unicharset file within language data

[16] It is not stated if text output re-ordering processing for *Parichit* recognition output was applied before accuracy was measured.

[17] https://github.com/yasserg/crawler4j

[18] https://github.com/kohlschutter/boilerpipe

[19] https://lucene.apache.org/core/

yalam dependent vowels found between consonants in consonant ligatures were not segmented.

Language	Letters
Khmer	◌ើ ◌ឿ ◌ៀ ◌េ ◌ែ ◌ៃ ◌ះ ◌ៈ [U+17BE - U+17C3, U+17C7, U+17C8] ◌ោ ◌ៅ (left components) [U+17C4 and U+17C5]
Malayalam	◌ം ◌ഃ ◌ാ ◌ി ◌ീ ◌ു ◌ൂ ◌ൃ ◌ൄ ◌െ ◌േ ◌ൈ ◌ൊ ◌ോ ◌ൌ ◌ൗ [U+0D02, U+0D03, U+0D3E - U+0D4C, U+0D57]
Odia	◌ଂ ◌ଃ ◌ା ◌ୀ ◌େ ◌ୈ ◌ୋ ◌ୌ [U+0B02, U+0B03, U+0B3E, U+0B40, U+0B47 - U+0B4C]

Table 4: Letters and consonant clusters affected by visual segmentation processing per language

The size of corpus and number of glyphs according to logical segmentation is given in Table 5.

Language	Text corpus (Mb)	Logical glyphs (million)
Khmer	252	137.0
Malayalam	307	134.8
Odia	68.9	96.6

Table 5: Text corpus size and occurrences of logical glyphs per language

3.1.2 Tesseract training data

Tesseract training data was prepared for each language using the paired sets of glyph data described in section 3.1. An application was implemented to automatically create Tesseract training data from each glyph data set, with the ability to automatically delete dotted consonant outlines displayed when a Unicode dependent letter or sign is rendered separately. The implemented application outputs multi-page TIFF format images and corresponding bounding box coordinates in the Tesseract training data format.[20]

Tesseract training was completed using most recent release v3.02 according to the documented training process for Tesseract v3, excluding shapeclustering. The number of examples of each glyph, between 5 and 40 in each training set, was determined by relative frequency in the corpus. A limited set of punctuation and symbols were also added to each set of glyph data, equal to those included in *tesseract-ocr* project language data. However, training text was not representative as recommended in documentation, with glyphs and punctuation randomly sorted.

3.1.3 Dictionary data

As dictionary data is utilised during Tesseract segmentation processing, word lists were prepared for each segmentation approach. As the separated character approach introduced a visual ordering to some consonant-vowel combinations and consonant clusters, word lists to be used in this approach were re-ordered, in line with the segmentation processing used for each language described in section 3.1. Word lists were extracted from the *tesseract-ocr* project v3.04 language data.

3.1.4 Ground truth data

OCR ground truth data was prepared in a single font size for each language in the PAGE XML format (Pletschacher and Antonacopoulos. 2010) using the application also described in section 3.1.2. The implementation segments text according to logical or visual ordering described in section 3.1.1, and uses the *Java PAGE libraries*[21] to output PAGE XML documents.

Text was randomly selected from documents within the web corpora described in section 3.1. Text segments written in Latin script were removed. Paired ground truth data were then generated. For each document image, two corresponding ground truth PAGE XML files were created according to logical and visual segmentation methods.

3.1.5 Evaluation

Tesseract v3.04 was used via the *Aletheia* v3 tool for production of PAGE XML ground truth described by Clausner et al. (2014). Evaluation was completed using the layout evaluation framework for evaluating PAGE XML format OCR outputs and ground truth described by Clausner et al. (2011). Output evaluations were completed using the described *Layout Evaluation* tool and stored in XML format.

[20] Description of the training format and requirements can be found at https://github.com/tesseract-ocr/tesseract/wiki/TrainingTesseract

[21] The PAGE XML format and related tools have been developed by the PRImA Research Lab at the University of Salford, and are available from http://www.primaresearch.org/tools/

3.2 Results

Results are presented in three sections; for *tesseract-ocr* language data, for web corpora glyph data per segmentation method, and for the comparable Tesseract language data per segmentation method.

Measured layout success is a region correspondence determination. Results are given for glyph based count and area weighted arithmetic and harmonic mean layout success as calculated by the *Layout Evaluation* tool. Weighted area measures are based on the assumption that bigger areas regions are more important than smaller ones, while the weighted count only takes into account the error quantity.

3.2.1 *Tesseract-ocr* language data

Recognition accuracy for selected *tesseract-ocr* project language data with Indic scripts is given in Table 6. All glyphs are segmented in line with Unicode logical encoding standards; using a logical segmentation approach, except for Thai and Lao which are encoded with visual segmentation in Unicode.

Measured Thai recognition accuracy is in line with the 79.7% accuracy reported by Smith (2014). While Hindi accuracy is far less than the 93.6% reported by Smith (2014), it is higher than the 73.3% found by Krishnan et al. (2014). Measured recognition accuracy for Telugu is also higher than the 67.1% found by Krishnan et al. (2014), although this may be expected for higher quality evaluation images. Measured Khmer recognition accuracy is in line with the 50-60% reported in Tan (2014). Bengali results are within the 70-93% range reported by Hasnat et al. (2009a), but are not directly comparable with the training approach used in *BanglaOCR*.

3.2.2 Web corpora glyphs by logical and visual segmentation

The number of glyphs and their occurrences in the collected language specific Wikipedia corpora are shown in Figure 4. These are compared to the number of glyphs in the *tesseract-ocr* project language data recognition character set[22], and the number of glyphs when visual order segmentation processing is applied to that character set. Visual segmentation can be seen to significantly reduce the number of glyphs for the same language coverage in each case. The logical glyphs in common and unique to *tesseract-ocr* and corpus based language data may be seen in Figure 3.

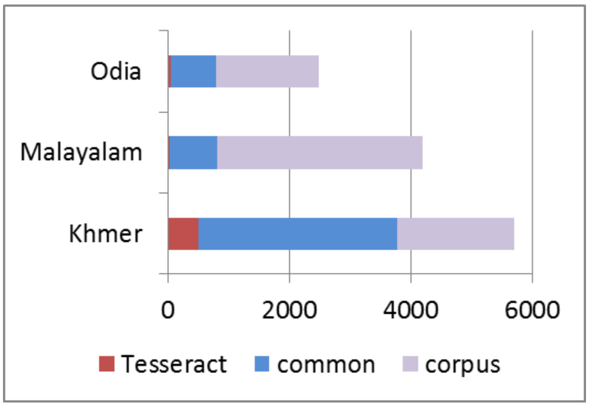


Figure 3: Coverage of logical glyphs between *tesseract-ocr* and corpus based language data

3.2.3 Comparable data for logical and visual segmentation

The total number of examples in the training data and size of the resulting Tesseract language data file with each approach (without dictionary data) is given in Table 7. The *tesseract-ocr* language data sizes are not directly comparable as the training sets and fonts differ.

OCR recognition accuracy is given for each segmentation method in Table 7. Recognition accuracy was found to be higher for visual segmentation in each language; by 3.5% for Khmer, 16.1% for Malayalam, and by 4.6% for Odia.

Logical segmentation accuracy shown in Table 7 was measured against the same ground truth data reported in section 3.2.1. However, as illustrated in Figure 4, the coverage of glyphs in each set of language data differed greatly. In each case, the number of glyphs found in the collected corpus was significantly greater than in the *tesseract-ocr* recognition set.

Recognition accuracy for *tesseract-ocr* language data for Khmer and Malayalam was 12.2% and 13% higher respectively than for the corpus based logical segmentation language data when measured against the same ground truth. However the corpus based logical segmentation data for Odia achieved 12.2% higher recognition accuracy than *tesseract-ocr* language data.

Dictionary data added to language data for each segmentation method was found to make no more than 0.5% difference to recognition or layout accuracy for either segmentation method.

[22] Glyphs not within the local language Unicode range(s) are not included.

Language	Recognition	Mean overall layout success (%)				Ground truth		Recognition
	accuracy	Area		Count		Glyphs	Char	glyphs
	(%)	weighted		weighted		(logical)		
		Arith.	Har.	Arith.	Har.			
Assamese	26.1	65.3	49.6	59.5	47.2	1080	1795	1506
Bengali	71.8	92.7	91.9	66.8	63.5	1064	1932	1451
Khmer	52.2	92.6	92.1	82.9	81.0	556	1099	3865
Lao *	77.1	96.6	96.5	85.6	84.1	1139	1445	1586
Gujarati	1.8	69.6	64.2	57.6	53.1	974	1729	1073
Hindi	81.9	89.1	87.4	58.2	49.4	952	1703	1729
Malayalam	62.7	90.6	89.2	82.5	78.1	552	1153	855
Myanmar	25.6	86.8	84.4	67.2	59.2	598	1251	7625
Odia	63.7	96.3	96.1	90.0	88.7	864	1514	834
Punjabi **	0.1	61.4	41.6	65.4	52.3	916	1569	1029
Tamil	89.2	95.5	95.0	93.1	92.4	798	1290	295
Telugu	75.3	78.0	72.6	55.1	44.2	877	1674	2845
Thai *	79.7	95.1	94.7	86.7	85.7	1416	1727	864

Table 6: Glyph recognition and layout accuracy for *tesseract-ocr* project v3.04 language data for selected Indic languages *languages encoded in visual segmentation in Unicode ** written in Gurmukhi script

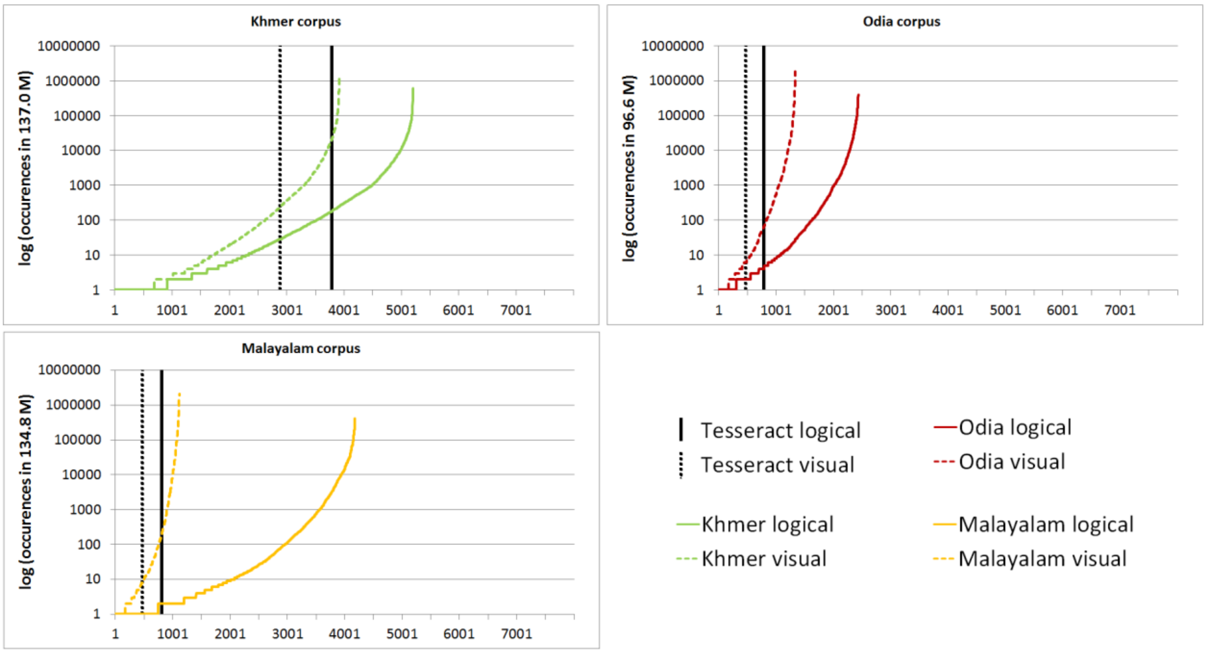


Figure 4: Comparison of logical vs. visual segmentation of glyphs in corpora

Language	Seg-	Recogni-	Mean overall layout success (%)				Ground	Recognition
	menta-	tion accu-	Area		Count		truth	glyphs
	tion	racy (%)	weighted		weighted		glyphs	
			Arith.	Har .	Arith.	Har.		
Khmer	Logical	41.0	92.8	91.9	83.6	80.5	556	5205
	Visual	44.5	92.9	92.3	86.9	85.8	677	3965
Malayalam	Logical	54.2	90.2	88.4	80.4	74.3	552	4237
	Visual	70.3	90.8	89.7	80.5	77.6	851	1171
Odia	Logical	75.9	94.8	94.4	88.2	86.4	864	2491
	Visual	80.5	95.1	94.7	91.5	90.8	1130	1387

Table 7: Glyph recognition and layout accuracy, ground truth and language data for logical and visual segmentation

4 Discussion

Analysis of the collected glyph corpora and *tesseract-ocr* project language data has shown the visual segmentation significantly reduces the number of glyphs required for a Tesseract training set in each of the languages considered. When using comparative training and ground truth data, visual segmentation was also shown to reduce the size of Tesseract language data and increase recognition accuracy. The use of dictionary data was not found to significantly affect results.

The implementation for visual segmentation of glyphs led to inconsistencies between similar visual components. For example, in Khmer it was observed that the visual representation of *coeng* (U+17D2) was commonly segmented by Tesseract as a separate glyph using *tesseract-ocr* and created language data, as illustrated for Khmer in Figure 5. Further opportunities for visual segmentation were also not implemented, such as components of consonant clusters. A consistent and more sophisticated implementation of visual segmentation may further improve results.

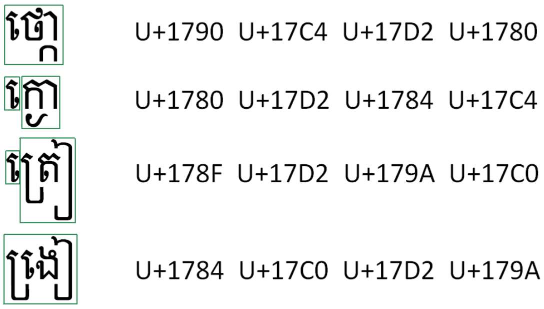

U+1790 U+17C4 U+17D2 U+1780
U+1780 U+17D2 U+1784 U+17C4
U+178F U+17D2 U+179A U+17C0
U+1784 U+17C0 U+17D2 U+179A

Figure 5: Visual glyphs for Khmer as implemented

The Tesseract training data prepared from corpus based glyphs was intended to be comparable, but was not in line with recommendations for training Tesseract. Preparation of training data in line with recommendations may improve results. The effects of Tesseract configuration parameters were not investigated during this study and should also be explored per language. Further, while glyph recognition accuracy achieved for the visual segmentation language data for Khmer was lower than that of the *tesseract-ocr* project language data, the coverage of glyphs was far greater. A significant percentage of the glyphs in each training set were rare. Future work may examine the relationship between coverage of rare glyphs in language data and recognition accuracy.

While effort was made to estimate coverage of modern glyphs for each segmentation approach in each language, the web corpora collected may not be representative. In preparing training data for the proposed segmentation method, care must be taken to determine that isolated or combined characters in the training sets are rendered in the predicted way when combined with other characters. A further consideration when creating multi-font training data is that characters may be rendered significantly differently between fonts. Further, some scripts have changed over time. For example, Malayalam has undergone formal revision in the 1970s, and informal changes with computer-aided typesetting in the 1980s, and Devanagari has also modified specific characters during the last three decades.

5 Conclusion

Developing high accuracy, multi-font language data for robust, end-to-end processing for Tesseract was not within the scope of this study. Rather, the aim was an initial investigation of alternate approaches for logical compared to visual character segmentation in a selection of Indic writing scripts. Results in the limited evaluation domain indicate that the proposed visual segmentation method improved results in three languages. The described technique may potentially be applied to further Indic writing scripts. While recognition accuracy achieved for the reported languages remains relatively low, outcomes indicate that effort to implement language specific training data preparation and OCR output reordering may be warranted.

Acknowledgements

We would like to acknowledge Google and the work of Mr. Ray Smith and contributors to the *tesseract-ocr* project.

We would also like to acknowledge the PRImA Lab of the University of Salford for their work in developing the PAGE XML format and related software tools and applications.

References

Chattopadhyay, T., Sinha, P., and Biswas, P. *Performance of document image OCR systems for recognizing video texts on embedded platform*. International Conference on Computational Intelligence and Communication Networks (CICN), 2011, pp. 606-610, Oct 2011

Clausner, C., Pletschacher, S. and Antonacopoulos, A. 2014. *Scenario Driven In-Depth Performance Evaluation of Document Layout Analysis Methods*. In proc. of the 11th International Conference on Document Analysis and Recognition (ICDAR2011), Beijing, China, September 2011, pp. 1404-1408

Clausner, C., Pletschacher, S. and Antonacopoulos, A. 2014. *Efficient OCR Training Data Generation with Aletheia*. Short paper booklet of the 11th International Association for Pattern Recognition Workshop on Document Analysis Systems (DAS2014), Tours, France, April 2014, pp. 19-20

de Does, Jesse. and Depuydt, Katrien. 2012. *Lexicon-supported OCR of eighteenth century Dutch books: a case study*. In proc. SPIE 8658, Document Recognition and Retrieval XX, 86580L (February 4, 2013); doi:10.1117/12.2008423

Dhiman and Singh. 2013. *Tesseract Vs Gocr A Comparative Study*. International Journal of Recent Technology and Engineering (IJRTE): Vol 2, Issue 4, September 2013

Govindaraju, Venugopal, and Srirangaraj Setlur. 2009. *Guide to OCR for Indic scripts: document recognition and retrieval.* London: Springer.

Hasnat, Abul., Chowdhury, Muttakinur Rahman. and Khan, Mumit. 2009a. *An open source Tesseract based Optical Character Recognizer for Bangla script*. In proc. Tenth International Conference on Document Analysis and Recognition (ICDAR2009), Catalina, Spain, July 26-29, 2009

Hasnat, Abul., Chowdhury, Muttakinur Rahman. and Khan, Mumit. 2009b. *Integrating Bangla script recognition support in Tesseract OCR*. In proc. Conference on Language Technology 2009 (CLT09), Lahore, Pakistan, January 22-24, 2009

Heliński, Marcin., Kmieciak, Miłosz. and Parkoła, Tomasz. 2012. *Report on the comparison of Tesseract and ABBYY FineReader OCR engines*. IMPACT Report. http://www.digitisation.eu/download/IMPACT_D-EXT2_Pilot_report_PSNC.pdf Last Accessed 3/9/2015

Ibrahim, Ahmed. 2014. *Dhivehi OCR: Character Recognition of Thaana Script using Machine Generated Text and Tesseract OCR Engine*, Edith Cowan University, Australia. http://www.villacollege.edu.mv/iri/images/thaana2.pdf, Last accessed 3/9/2015

Krishnan, Praveen., Sankaran, Naveen., Singh, and Ajeet Kumar. 2014. *Towards a Robust OCR System for Indic Scripts*, 11th IAPR International Workshop on Document Analysis Systems (DAS2014), Tours, France, 7th – 10th April 2014.

Mishra, Nitin; Patvardhan, C.; Lakshimi, Vasantha C.; and Singh, Sarika. 2012. *Shirorekha Chopping Integrated Tesseract OCR Engine for Enhanced Hindi Language Recognition*, International Journal of Computer Applications, Vol. 39, No. 6, February 2012, pp. 19-23

Nishad, A; and Bindu, K. 2013. *Malayalam OCR Systems – A Survey*. International Journal of Computer Technology and Electronics Engineering, Vol. 3, No. 6, December 2013

Nayak, Mamata. and Nayak, Ajit Kumar. 2014. *Odia Characters Recognition by Training Tesseract OCR Engine*. In proc. International Conference in Distributed Computing and Internet Technology 2014 (ICDCIT-2014)

Omee, Farjana Yeasmin., Himel, Shiam Shabbir. and Bikas, Md. Abu Naser. 2011. *A Complete Workflow for Development of Bangla OCR*. International Journal of Computer Applications, Vol. 21, No. 9, May 2011

Patel, Chirag., Patel, Atul and Patel, Dharmendra. 2012. *Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study*, International Journal of Computer Applications, Vol. 55, No. 10

Pletschacher, S. and Antonacopoulos, A. 2010. *The PAGE (Page Analysis and Ground-Truth Elements) Format Framework*. In proc. of the 20th International Conference on Pattern Recognition (ICPR2010), Istanbul, Turkey, August 23-26, 2010, IEEE-CS Press, pp. 257-260

Pujari, Pushpalata; and Majhi, Babita. 2015. *A Survey on Odia Character Recognition*. International Journal of Emerging Science and Engineering (IJESE), Vol. 3, No. 4, February 2015

Rice, S.V., Jenkins, F.R. and Nartker, T.A. 1995. *The Fourth Annual Test of OCR Accuracy*, Technical Report 95-03, Information Science Research Institute, University of Nevada, Las Vegas

Smith, Ray D. Antonova and D. Lee. 2009 *Adapting the Tesseract Open Source OCR Engine for Multilingual OCR*, in proc. International Workshop on Multilingual OCR 2009, Barcelona, Spain, July 25, 2009

Smith, Ray. 2007. *An overview of the Tesseract OCR Engine*, in proc. Of the 9thInternational Conference on Document Analysis and Recognition (ICDAR2007), Curitiba, Paraná, Brazil, 2007

Smith, Ray. 2014. *Everything you always wanted to know about Tesseract.* 11th IAPR International Workshop on Document Analysis Systems (DAS2014), Tours, France, 7th – 10th April 2014. Tutorial slides available from https://drive.google.com/file/d/0B7l10Bj_LprhbUlIUFlCdGtDYkE/view?pli=1 Last visited 11/9/2015

Sok, Pongsametry. and Taing, Nguonly. 2014. *Support Vector Machine (SVM) Based Classifier For Khmer Printed Character-set Recognition*, Asia Pacific Signal and Information Processing Association, 2014, Annual Summit and Conference (APSIPA), 9-12 December, 2014, Siem Reap, city of Ankor Wat, Cambodia

Tan, Germaine. 2014. Khmer OCR: Convert Hard-Copy Khmer Text To Digital. Geeks in Cambodia, November 18, 2014. http://geeksincambodia.com/khmer-ocr-convert-hard-copy-khmer-text-to-digital/ Last visited 13/9/2015

Vijayarani and Sakila. 2015. *Performance Comparison of OCR Tools*. International Journal of UbiComp (IJU), Vol. 6, No. 3, July 2015

Vincent, Luc. 2006. *Announcing Tesseract OCR*, Google Code, http://googlecode.blogspot.com.au/2006/08/announcing-tesseract-ocr.html Last accessed 1/9/2015

White, Nick. 2013. *Training Tesseract for Ancient Greek OCR*. The Eutypon, No. 28-29, October 2013, pp. 1-11. http://ancientgreekocr.org/e29-a01.pdf Last visited 18/9/2015

Yadav, Divakar; Sanchez-Cuadrado, Sonia; and Morato, Jorge. 2013. *Optical Character Recognition for Hindi Language Using a Neural Network Approach*. Journal of Information Processing Systems, Vol. 9, No. 10, March 2013, pp. 117-140

Analysis of Word Embeddings and Sequence Features for Clinical Information Extraction

Lance De Vine, Mahnoosh Kholghi, Guido Zuccon, Laurianne Sitbon
Queensland University of Technology
{l.devine, m1.kholghi, g.zuccon, laurianne.sitbon}@qut.edu.au
Anthony Nguyen
The Australian e-Health Research Centre, CSIRO
anthony.nguyen@csiro.au

Abstract

This study investigates the use of unsupervised features derived from word embedding approaches and novel sequence representation approaches for improving clinical information extraction systems. Our results corroborate previous findings that indicate that the use of word embeddings significantly improve the effectiveness of concept extraction models; however, we further determine the influence that the corpora used to generate such features have. We also demonstrate the promise of sequence-based unsupervised features for further improving concept extraction.

1 Introduction

Clinical concept extraction involves the identification of sequences of terms which express meaningful concepts in a clinical setting. The identification of such concepts is important for enabling secondary usage of reports of patient treatments and interventions, e.g., in the context of cancer monitoring and reporting (Koopman et al., 2015), and for further processing in downstream eHealth workflows (Demner-Fushman et al., 2009).

A significant challenge is the identification of concepts that are referred to in ways not captured within current lexical resources such as relevant domain terminologies like SNOMED CT. Furthermore, clinical language is sensitive to ambiguity, polysemy, synonymy (including acronyms) and word order variations. Finally, the information presented in clinical narratives is often unstructured, ungrammatical, and fragmented.

State of the art approaches in concept extraction from free-text clinical narratives extensively apply supervised machine learning approaches. The effectiveness of such approaches generally depends on three main factors: (1) the availability of a considerable amount of high quality annotated data, (2) the selected learning algorithm, and (3) the quality of features generated from the data.

In recent years, clinical information extraction and retrieval challenges like i2b2 (Uzuner et al., 2011) and ShARe/CLEF (Suominen et al., 2013) have provided annotated data which can be used to apply and evaluate different machine learning approaches (e.g., supervised and semi-supervised). Conditional Random Fields (CRFs) (Lafferty et al., 2001) has shown to be the state-of-the-art supervised machine learning approach for this clinical task. A wide range of features has been leveraged to improve the effectiveness of concept extraction systems, including hand-crafted grammatical, syntactic, lexical, morphological and orthographical features (de Bruijn et al., 2011; Tang et al., 2013), as well as advanced semantic features from external resources and domain knowledge (Kholghi et al., 2015).

While there has been some recent work in the application of unsupervised machine learning methods to clinical concept extraction (Jonnalagadda et al., 2012; Tang et al., 2013), the predominant class of features that are used are still handcrafted features.

This paper discusses the application to clinical concept extraction of a specific unsupervised machine learning method, called the Skip-gram Neural Language Model, combined with a lexical string encoding approach and sequence features. Skip-gram word embeddings, where words are represented as vectors in a high dimensional vector space, have been used in prior work to create feature representations for classification and information extraction tasks, e.g., see Nikfarjam et al. (2015) and Qu et al. (2015). The following research questions will be addressed in this paper:

RQ1: are word embeddings and sequence level representation features useful when using CRFs for clinical concept extraction?

RQ2: to what extent do the corpora used to gener-

Lance De Vine, Mahnoosh Kholghi, Guido Zuccon, Laurianne Sitbon and Anthony Nguyen. 2015. Analysis of Word Embeddings and Sequence Features for Clinical Information Extraction . In *Proceedings of Australasian Language Technology Association Workshop*, pages 21−30.

ate such unsupervised features influence the effectiveness?

Question one has been partially addressed by prior work that has shown word embeddings improve the effectiveness of information extraction systems (Tang et al., 2015; Nikfarjam et al., 2015). However, we further explore this by considering the effectiveness of sequence level features, which, to the best of our knowledge, have not been investigated in clinical information extraction.

2 Related Work

The two primary areas that relate to this work include *(a)* methods for clinical concept extraction, and *(b)* general corpus based approaches for learning word representations.

2.1 Clinical Information Extraction

The strong need for effective clinical information extraction methods has encouraged the development of shared datasets such as the i2b2 challenges (Uzuner et al., 2011) and the ShARe/CLEF eHealth Evaluation Lab (Suominen et al., 2013); which in turn have sparked the development of novel, more effective clinical information extraction methods. For example, de Bruijn et al. (2011) used token, context, sentence, section, document, and concept mapping features, along with the extraction of clustering-based word representation features using Brown clustering; they obtained the highest effectiveness in the i2b2/VA 2010 NLP challenge. In the same challenge, Jonnalagadda et al. (2012) leveraged distributional semantic features along with traditional features (dictionary/pattern matching, POS tags). They used random indexing to construct a vector-based similarity model and observed significant improvements.

Tang et al. (2013) built a concept extraction system for ShARe/CLEF 2013 Task 1 that recognizes disorder mentions in clinical free text, achieving the highest effectiveness amongst systems in the challenge. They used word representations from Brown clustering and random indexing, in addition to a set of common features including token, POS tags, type of notes, section information, and the semantic categories of words based on UMLS, MetaMap, and cTAKEs.

Tang et al. (2014) extracted two different types of word representation features: (1) clustering-based representations using Brown clustering, and (2) distributional word representations using random indexing. Their findings suggest that these word representation features increase the effectiveness of clinical information extraction systems when combined with basic features, and that the two investigated distributional word representation features are complementary.

Tang et al. (2014), Khabsa and Giles (2015) and Tang et al. (2015) investigated the effect of three different types of word representation features, including clustering-based, distributional and word embeddings, on biomedical name entity recognition tasks. All developed systems demonstrated the significant role of word representations in achieving high effectiveness.

2.2 Corpus Based Methods for Word Representations

Brown clustering (Brown et al., 1992) has probably been the most widely used unsupervised method for feature generation for concept extraction. Both random indexing (Kanerva et al., 2000) and word embeddings from neural language models, e.g., Mikolov et al. (2013), have also been used recently, in part stimulated by renewed interest in representation learning and deep learning. Some of the more notable contributions to the use of word representations in NLP include the work of Turian et al. (2010) and Collobert et al. (2011). Since their inception, Skip-gram word embeddings (Mikolov et al., 2013) have been used in a wide range of settings, including for unsupervised feature generation (Tang et al., 2015). There have also been recent applications of convolutional neural nets to lexical representation. For example, Zhang and LeCun (2015) demonstrated that deep learning can be applied to text understanding from character-level inputs all the way up to abstract text concepts, using convolutional networks.

3 Features

We start by examining a set of baseline features that have been derived from previous work in this area. We then turn our attention to unsupervised features to be used in this task and we propose to examine features based on word embeddings, lexical vectors and sequence level vectors. These features will then be tested to inform a CRFs learning algorithm, see Figure 1.

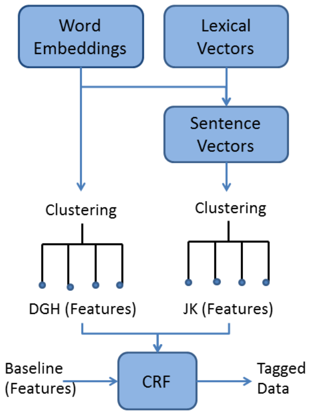


Figure 1: Feature generation process and their use in concept extraction.

3.1 Baseline Features

We construct a baseline system using the following baseline feature groups, as described by Kholghi et al. (2015):

A: Orthographical (regular expression patterns), lexical and morphological (suffixes/prefixes and character n-grams), contextual (window of k words),

B: Linguistic (POS tags (Toutanova et al., 2003))

C: External resource features (UMLS and SNOMED CT semantic groups as described by Kholghi et al. (2015)).

3.2 Unsupervised Features

The approach we use for generating unsupervised features consists of the following two steps:

1. Construct real valued vectors according to a variety of different methods, each described in Sections 3.2.1– 3.2.3.
2. Transform the vectors into discrete classes via clustering, as described in Section 3.2.4.

While real valued feature vectors can be used directly with some CRFs software implementations, they are not supported by all. We have found that transforming our vectors into discrete classes via clustering is reasonably easy. In addition our preliminary experiments did not show advantages to working with real valued vectors.

We use two types of vectors: semantic and lexical. We use the term "semantic" as an overarching term to refer to neural word embeddings as well as other distributional semantic representations such as those derived from random indexing. The semantic vectors encode a combination of semantic and syntactic information, as distinct to lexical vectors which encode information about the distribution of character patterns within tokens. We find that lexical vectors identify lexical classes within a corpus and are particular useful for corpora where there are many diverse syntactic conventions such as is the case with clinical text.

3.2.1 Semantic Vectors

To construct semantic vectors we use the recently proposed Skip-gram word embeddings. The Skip-gram model (Mikolov et al., 2013) constructs term representations by optimising their ability to predict the representations of surrounding terms.

Given a sequence $\mathcal{W} = \{w_1, \dots, w_t, \dots, w_n\}$ of training words, the objective of the Skip-gram model is to maximise the average log probability

$$\frac{1}{2r} \sum_{i=1}^{2r} \sum_{-r \leq j \leq r, j \neq 0} \log p(w_{t+j}|w_t) \qquad (1)$$

where r is the context window radius. The context window determines which words are considered for the computation of the probability, which is computed according to

$$p(w_O|w_I) = \frac{\exp(v_{w_O}^{\top} v_{w_I})}{\sum_{w=1}^{W} \exp({v_w}^{\top} v_{w_I})} \qquad (2)$$

where the v_{w_I} and v_{w_O} are vector representations of the input and output (predicted) words. The value (2) is a normalized probability because of the normalization factor $\sum_{w=1}^{W} \exp(v_w^{\top} v_{w_I})$. In practice, a hierarchical approximation to this probability is used to reduce computational complexity (Morin and Bengio, 2005; Mikolov et al., 2013).

At initialisation, the vector representations of the words are assigned random values; these vector representations are then optimised using gradient descent with decaying learning rate by iterating over sentences observed in the training corpus.

3.2.2 Lexical Vectors

Various approaches have been previously used to encode lexical information in a distributed vector representation. A common idea in these approaches is the hashing and accumulation of n-grams into a single vector. This is sometimes referred to as string encoding and is used in a variety of applications, including text analysis and bio-informatics (Buhler, 2001; Buckingham et al., 2014). The approach used here is most similar to the holographic word encoding approach of

Hannagan et al. (2011) and Widdows and Cohen (2014).

To create lexical vectors, we first generate and associate a random vector for each distinct character n-gram that is found in the text. Then, for each token we accumulate the vectors for each n-gram contained within the token. We use uni-grams, bi-grams, tri-grams and tetra-grams, but we also include skip-grams such as the character sequence "a_b" where the underscore is a wild-card placeholder symbol. The n-gram vectors are added together and the resulting vector is normalized.

Lexical feature representation is especially useful when there doesn't exist an easily available semantic representation. Some corpora, such as clinical texts, use an abundance of syntactic conventions, such as abbreviations, acronyms, times, dates and identifiers. These tokens may be represented using a lexical vector such that orthographically similar tokens will have similar vectors. An advantage of the use of these lexical vectors is that they are constructed in a completely unsupervised fashion which is corpus independent and does not rely on the use of hand-crafted rules. This is useful in the application to unseen data where there may exist tokens or patterns that have not been seen within the training set (which would in turn render most hand-crafted rules ineffective).

3.2.3 Sequence Level Vectors

Many models of phrase and sentence representation have recently been proposed for tasks such as paraphrase identification, sentiment classification and question answering (Le and Mikolov, 2014; Kalchbrenner et al., 2014), just to name a few. The simple approach adopted in this paper makes use of both semantic and lexical vectors.

To form sequence level vectors, we accumulate the word embeddings for each token in a phrase or sentence. A token is ignored if it does not have an associated word embedding. The lexical vectors for each token in a sequence are also accumulated. Both types of vectors, semantic and lexical, are normalized. We then concatenate the vectors and normalize again.

From time to time, some of the tokens within short text sequences may not be associated to word embeddings. In such a case the sequence is represented entirely with its accumulated lexical vectors. In this paper we evaluate the effectiveness of sentence and bi-gram phrase vectors.

3.2.4 Clustering Methodology

In our approach, the real valued vector representations obtained employing the methods above are then transformed into discrete classes. To cluster these vectors, we use K-means++ (Arthur and Vassilvitskii, 2007) with Euclidean distance using a range of different granularities akin to how multiple levels of representations are generally used in Brown clustering.

Clustering of vectors is performed on a training dataset. When a model is applied to unseen data, the representation for an unseen item is projected into the nearest cluster obtained from the training data, and a feature value is assigned to the item. We experimented with different strategies for assigning feature identifiers to clusters including (a) a simple enumeration of clusters, and (b) a reduced feature space in which only clusters containing a majority of members with the same configuration of concept labels (from training data) are given an incrementing feature number. Method (b) did not improve results and so we only report the outcomes of method (a). Clustering iterations were terminated at 120 iterations. Table 1 and 2 show examples of word and sentence clusters obtained from a clinical corpus.

4 Experimental Setup

To evaluate the feature groups studied in this paper, we use the annotated train and test sets of the i2b2/VA 2010 NLP challenge (Uzuner et al., 2011). We evaluate the effectiveness of concept extraction systems using Precision, Recall and F1-measure. Evaluation measures are computed on the i2b2 test data using MALLET's multi-segmentation evaluator (McCallum, 2002) as per the experimental setup of (Kholghi et al., 2014).

We compute statistical significance (p-value) using a 5*2 cross validated t-test (Dietterich, 1998) in which we combine both train and test

Table 1: Example of word embedding clusters.

C_1	prediabetes, insulin-dependant, endocrine., early-onset, type-2
C_2	flank/right, extremity/lower, mid-to-lower, extremity/right
C_3	knife, scissors, scalpel, clamp, tourniquet
C_4	instructed, attempted, allowed, refuses, urged
C_5	psychosomatic, attention-deficit, delirium/dementia, depression/bipolar

Table 2: Example of sentence clusters.

C_1	Abs Eos , auto 0.1 X10E+09/L ABS Lymphs 2.4 X10E+09 / L ABS Monocytes 1.3 X10E+09 / L Abs Eos , auto 0.2 X10E+09 / L
C_2	5. Dilaudid 4 mg Tablet Sig : ... 7. Clonidine 0.2 mg Tablet Sig : ... 9. Nifedipine 30 mg Tablet Sustained ... 10. Pantoprazole 40 mg Tablet ...
C_3	Right proximal humeral fracture status ... Bilateral renal artery stenosis status ... status post bilateral knee replacement ...

sets, sample 5 subsets of 30,000 sentences, split each subset into train and test, and perform a paired t-test for these 10 subsets.

As supervised machine learning algorithm for concept extraction, we used a linear-chain CRFs model based on the MALLET CRFs implementation and tuned following Kholghi et al. (2014). We use our own implementation of K-means++ for clustering. For creating the Skip-gram word embeddings we use the popular `word2vec` tool (Mikolov et al., 2013), with hierarchical softmax and 5 epochs on the C1 and C2 datasets and 1 epochs on the PM and WK datasets (see below) due to computational constrains.

4.1 Corpora

We use four different corpora to generate word embeddings[1]: two clinical (C1 and C2) and two non-clinical (PM and WK); corpora details are reported below and in Table 3:

C1: (Clinical) composed by the concatenation of the i2b2 train set (Uzuner et al., 2011), MedTrack (Voorhees and Tong, 2011), and the CLEF 2013 train and test sets (Suominen et al., 2013)

C2: (Clinical) the i2b2 train set (Uzuner et al., 2011)

PM: (Biomedical) PubMed, as in the 2012 dump[2]

WK: (Generalist) Wikipedia, as in the 2009 dump (De Vries et al., 2011)

4.2 Feature Groups

In addition to the feature groups `A`, `B` and `C` mentioned in Section 3.1, we consider the following feature groups:

Table 3: Training corpora for word embeddings.

Corpus	Vocab	Num. Tokens
C1	104,743	$\approx$ 29.5 M
C2	11,727	$\approx$ 221.1 K
PM	163,744	$\approx$ 1.8 B
WK	122,750	$\approx$ 415.7 M

`D`: Skip-gram clustering features with window size 2 and 5 and 128, 256, 512, 1024 clusters

`G`: Window of 3 previous and next Skip-gram clustering feature (window size 2) with 1024 clusters

`H`: Window of 3 previous and next Skip-gram clustering feature (window size 5) with 1024 clusters

`J`: Sentence features with 1024 clusters

`K`: Sentence features with 256 clusters

`L`: Bi-gram phrase features with 512 clusters

`M`: Bi-gram phrase features with 1024 clusters

5 Results and Discussion

In this section, we first study the impact of different feature sets on the effectiveness of the learnt models. We then discuss how different training corpora affect the quality of word embeddings and sequence representations.

5.1 Analysis of Baseline Features

Table 4 reports the effectiveness of CRF models built using only the word tokens appearing in the documents (`Word`), and this feature along with different combinations of baseline features (`A`, `B`, `C`). These results show that feature group `A` (orthographical, lexical, morphological, and contextual features) provides significantly higher effectiveness compared to other individual feature groups. Semantic features (group `C`) also achieve reasonably high effectiveness compared to the use of `Word` features alone. However, POS tags (group `B`) provide inferior effectiveness. Indeed, when feature group `B` is used in combination with either `A` or `C`, no significant differences are observed compared to using `A` or `C` alone: POS tags do not improve effectiveness when combined with another, single feature group. It is the combination of all baseline features (`ABC`), instead, that provides the highest effectiveness.

[1] Pre-processing involving lower-casing and substitution of matching regular expressions was performed.

[2] `http://mbr.nlm.nih.gov/Download/`

Table 4: Results for baseline features. Statistically significant improvements ($p<0.05$) for F1 when compared with Word are indicated by *.

Feature Set	Precision	Recall	F1
Word	0.6571	0.6011	0.6279
A	0.8404	0.8031	0.8213
B	0.6167	0.6006	0.6085
C	0.7691	0.6726	0.7192
BC	0.7269	0.712	0.7194
AB	0.8368	0.8038	0.8200
AC	0.8378	0.8059	0.8216
ABC	**0.8409**	**0.8066**	**0.8234***

Table 5: Results for word embedding features. The highest effectiveness obtained by each feature group is highlighted in bold. Statistically significant improvements ($p<0.05$) for F1 when compared with ABC are indicated by *.

Features	Corp	Prec.	Recall	F1
D	C1	0.7758	**0.7392**	**0.7571**
	C2	0.7612	0.6926	0.7252
	PM	**0.7776**	0.7309	0.7535
	WK	0.733	0.6534	0.6909
GH	C1	0.7868	**0.7469**	0.7663
	C2	0.7847	0.7001	0.7400
	PM	**0.8005**	0.7466	**0.7726**
	WK	0.7106	0.6043	0.6532
ABCD	C1	0.8432	0.8123	**0.8275**
	C2	**0.8435**	0.8006	0.8215
	PM	0.8377	**0.8126**	0.8249
	WK	0.8409	0.8108	0.8256
ABCD GH	C1	**0.8509**	**0.8118**	**0.8309***
	C2	0.8386	0.8001	0.8189
	PM	0.8484	0.8088	0.8281
	WK	0.8397	0.8063	0.8226

5.2 Analysis of Word Embedding Features

We study the effect of word embeddings on concept extraction to answer our RQ1 (see Section 1). To do so, we select the best combination of baseline features (ABC) and measure the effectiveness of adding semantic and lexical vectors features (groups D, G, and H). Results are reported in Table 5.

The effectiveness of the derived information extraction systems is influenced by the training corpus used to produce the embeddings. Thus, the results in Table 5 are reported with respect to the corpora; the effect training corpora have on effectiveness will be discussed in Section 5.4.

The effectiveness obtained when using the word embedding features alone[3] (group D) is comparable to that observed when using baseline semantic features (group C, Table 4). Group D includes 8 clustering features with window sizes 2 and 5. When using features of the three words preceding and following the target word with 1024 clusters (groups G and H), higher effectiveness is observed, irrespectively of the corpus (apart from WK).

Further improvements are obtained when clustering features are used in conjunction with the baseline features. The improvements in effectiveness observed when adding both D and contextual word embedding clustering features (G and H) are statistically significant compared to feature groups ABC. These results confirm those found in previous work that explored the use of word embeddings to improve effectiveness in information extraction tasks, e.g., Tang et al. (2015).

Note that we did study the effectiveness of using feature groups G and H with different number of clusters (i.e., 128, 256, 512 and 1024); however, the highest effectiveness was achieved when considering 1024 clusters. Similarly, we also experimented with different settings of word embedding's window size and dimensionality; the results of these experiments are not included in this paper for brevity[4]. The outcome of these trials was that embeddings with window size 5 usually perform better than window size 2, though not significantly; however the highest effectiveness is achieved when both sizes 2 and 5 are used. We also observed that there are no significant differences between the effectiveness of learnt models using embeddings generated with 300 dimensions as opposed to 100. However, larger embeddings are computationally more costly than smaller ones (both in terms of computer clocks and memory). Therefore, in this paper, all results were produced using embeddings of dimension 100.

5.3 Analysis of Sequence Features

We also study the effect of sequence features on concept extraction to answer our RQ1. For this we select the best combination of baseline and word embedding features (ABCDGH) and measure the effectiveness of adding sequence features (groups

[3] In the following, when referring to using a feature group alone, we mean using that feature group, along with the target word string.

[4] But can be found as an online appendix at https://github.com/ldevine/SeqLab.

Table 6: Results for sequence features. The highest effectiveness obtained by each feature group is highlighted in bold. Statistically significant improvements ($p<0.05$) for F1 when compared with ABC are indicated by *.

Features	Corp	Prec.	Recall	F1
J	C1	0.6832	0.6693	0.6762
	C2	0.5926	0.6036	0.7012
	PM	**0.7408**	**0.6701**	**0.7037**
	WK	0.733	0.6534	0.6909
K	C1	**0.7646**	**0.6747**	**0.7169**
	C2	0.7241	0.6639	0.6927
	PM	0.735	0.6641	0.6978
	WK	0.7237	0.6609	0.6909
ABCD GHJ	C1	**0.8493**	**0.8136**	**0.8311**
	C2	0.8463	0.7968	0.8208
	PM	0.8475	0.8134	0.8301
	WK	0.8388	0.8087	0.8235
ABCD GHK	C1	0.8473	0.8066	**0.8265**
	C2	**0.8494**	0.7941	0.8208
	PM	0.8423	0.8061	0.8238
	WK	0.8399	**0.8103**	0.8249
ABCD GHJK	C1	0.8488	**0.8152**	**0.8316***
	C2	**0.8491**	0.7959	0.8216
	PM	0.8472	0.8151	0.8308
	WK	0.8364	0.8034	0.8195
L	C1	0.7601	**0.6763**	**0.7157**
	C2	0.7311	0.6014	0.6599
	PM	**0.7624**	0.6720	0.7144
	WK	0.7619	0.6646	0.7099
M	C1	0.7584	**0.6761**	**0.7148**
	C2	0.6456	0.6521	0.6488
	PM	**0.7602**	0.6725	0.7137
	WK	0.6588	0.6424	0.6505
ABCD GHJKL	C1	**0.8484**	0.8103	0.8289
	C2	0.8460	0.7931	0.8187
	PM	0.8444	**0.8147**	**0.8293***
	WK	0.8388	0.8024	0.8202
ABCD GHJKM	C1	**0.8505**	0.8144	**0.8320***
	C2	0.8457	0.7967	0.8205
	PM	0.8468	**0.8160**	0.8311
	WK	0.8306	0.8060	0.8181
ABCD GHJKLM	C1	**0.8504**	0.8116	0.8305*
	C2	0.8465	0.7959	0.8204
	PM	0.8477	**0.8152**	**0.8311***
	WK	0.8391	0.8028	0.8205

J, K (sentence) and L, M (phrase)). Results are reported in Table 6.

The use of either feature groups J, K, L, M alone provide results that are comparable to the baseline semantic feature (C) or the embedding features (D), but are less effective than the use of the previous combination of features (ABCDGH).

Adding sentence features J and K separately to the remaining feature groups shows mixed results with no significant changes compared to ABCDGH. Specifically, feature group J provides small improvements across different corpora, while insignificant decrease is observed on C1 and PM with feature group K. Similar results are obtained with L and M (not reported).

However, when we combine all sentence features together (ABCDGHJK) we observe small improvements across all corpora except WK. This suggests that the results are somewhat sensitive to variation in the corpora used to learn word embeddings and sequence representations – we explore this further in the next section.

When the phrase features are added to word embedding and sentence features, small improvements are observed both over word embeddings (ABCDGH) and word embeddings with sentence features (ABCDGHJK).

In summary, sequence features provide small, additional improvements over word embedding features in the task of clinical concept extraction (when clinical and biomedical corpora are used to learn sequence representations). Given the differences between word embeddings, sentence features and phrase features, the results suggest that perhaps phrase, rather than sentence level representations should be further explored.

5.4 Analysis of Training Corpora

The results obtained when employing embedding features (D, G, H) and sequence features (J, K, L, M) are influenced by the corpora used to compute the embeddings (see Table 5 and 6). We therefore address our RQ2: how sensitive are the features to the training corpora?

The empirical results suggest that using a small corpus such as i2b2 (C2) to build the representations does not provide the best effectiveness, despite the test set used for evaluation contains data that is highly comparable with that in C2 (this corpus contains only i2b2's train set). However, the highest effectiveness is achieved when augmenting C2 with data from clinical corpora like Medtrack and ShARe/CLEF (C1).

The results when PubMed (PM) is used to derive the feature representations are generally lower

Table 7: Number of target tokens contained in the i2b2 test set but not in each of the word embedding training corpora.

Corp	# Miss. Tok.	Corp	# Miss. Tok.
C1	196	PM	549
C2	890	WK	1152

but comparable to those obtained on the larger clinical corpus (C1) and always better than those obtained on the smaller clinical corpus (C2) and the Wikipedia data (WK).

Learning word embedding and sequence features from Wikipedia, in combination with the baseline features (i.e., ABCDGH and ABCDGHJKLM), results in (small) losses of effectiveness compared to the use of baseline features only (ABC), despite Wikipedia being one of the largest corpora among those experimented with. We advance two hypotheses to explain this: (1) Wikipedia contains less of the tokens that appear in the i2b2 test set than any other corpora (*poor coverage*), (2) for the test tokens that do appear in Wikipedia, word embedding representations as good as those obtained from medical data cannot be constructed because of the sparsity of domain aligned data (*sparse domain data*). The first hypothesis is supported by Table 7, where we report the number of target tokens contained in the i2b2 test dataset but not in each of the word embedding training corpora. The second hypothesis is supported by a manual analysis of the embeddings from WK and compared e.g. to those reported for C1 in Table 1. Indeed, we observe that embeddings and clusters in C1 address words that are misspelled or abbreviated, a common finding in clinical text; while, the representations derived from WK miss this characteristic (see also Nothman et al. (2009)). We also observe that the predominant word senses captured by many word vectors is different between medical corpora and Wikipedia, e.g., *episodes:* {*bouts, emesis, recurrences, ...*} in C1, while *episodes:* {*sequels, airings, series, ...*} in WK.

These results can be summarised into the following observations:

- C2 does not provide adequate coverage of the target test tokens because of the limited amount of data, despite its clinical nature;
- when using medical corpora, the amount of data, rather than its format or domain, is often more important for generating representations conducive of competitive effectiveness;
- data containing biomedical content rather than clinical content can be used in place of clinical data for producing the studied feature representations without experiencing considerable loss in effectiveness. This is particularly important because large clinical datasets are expensive to compile and are often a well guarded, sensitive data source;
- if content, format and domain of the data used to derive these unsupervised features is too different from that of the target corpus requiring annotations, then the features are less likely to deliver effective concept extraction.

6 Conclusions and Future Work

This paper has investigated the use of unsupervised methods to generate semantic and lexical vectors, along with sequence features for improving clinical information extraction. Specifically, we studied the effectiveness of these features and their sensitivity to the corpus used to generate them. The empirical results have highlighted that:

1. word embeddings improve information extraction effectiveness over a wide set of baseline features;
2. sequence features improve results over both baseline features (significantly) and embeddings features (to a less remarkable extent);
3. the corpora used to generate the unsupervised features influence their effectiveness, and larger clinical or biomedical corpora are conducive of higher effectiveness than small clinical corpora or large generalist corpora. These observations may be of guidance to others.

This study opens up a number of directions for future work. Other approaches to create lexical vectors exits, e.g., morpheme embeddings (Luong et al., 2013), or convolutional neural nets applied at the character level (Zhang and LeCun, 2015), and their effectiveness in this context is yet to be studied. Similarly, we only investigated an initial (but novel) approach to forming sequence representations for feature generation. Given the promise expressed by this approach, more analysis is required to reach firm conclusions about the effectiveness of sequence features (both sentence and phrase), including the investigation of alternative approaches for generating these feature groups.

References

David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Lawrence Buckingham, James M Hogan, Shlomo Geva, and Wayne Kelly. 2014. Locality-sensitive hashing for protein classification. In *Conferences in Research and Practice in Information Technology*, volume 158. Australian Computer Society, Inc.

Jeremy Buhler. 2001. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, 17(5):419–428.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. 2011. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association*, 18(5):557–562.

Christopher M De Vries, Richi Nayak, Sangeetha Kutty, Shlomo Geva, and Andrea Tagarelli. 2011. Overview of the inex 2010 xml mining track: Clustering and classification of xml documents. In *Comparative evaluation of focused retrieval*, pages 363–376. Springer.

Dina Demner-Fushman, Wendy W Chapman, and Clement J McDonald. 2009. What can natural language processing do for clinical decision support? *Journal of biomedical informatics*, 42(5):760–772.

Thomas G Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Thomas Hannagan, Emmanuel Dupoux, and Anne Christophe. 2011. Holographic string encoding. *Cognitive Science*, 35(1):79–118.

Siddhartha Jonnalagadda, Trevor Cohen, Stephen Wu, and Graciela Gonzalez. 2012. Enhancing clinical concept extraction with distributional semantics. *Journal of biomedical informatics*, 45(1):129–140.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Pentti Kanerva, Jan Kristofersson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd annual conference of the cognitive science society*, volume 1036.

Madian Khabsa and C Lee Giles. 2015. Chemical entity extraction using crf and an ensemble of extractors. *J Cheminform*, 7(Suppl 1):S12.

Mahnoosh Kholghi, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. 2014. Factors influencing robustness and effectiveness of conditional random fields in active learning frameworks. In *Proceedings of the 12th Australasian Data Mining Conference*, AusDM'14. Australian Computer Society.

Mahnoosh Kholghi, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. 2015. External knowledge and query strategies in active learning: A study in clinical information extraction. In *Proceedings of the 24rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '15, New York, NY, USA. ACM.

Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. 2015. Automatic ICD-10 classification of cancers from free-text death certificates. *International Journal of Medical Informatics*, 84(11):956 – 965.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. *CoNLL-2013*, 104.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246–252.

Azadeh Nikfarjam, Abeed Sarker, Karen O'Connor, Rachel Ginn, and Graciela Gonzalez. 2015. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the*

American Medical Informatics Association, page ocu041.

Joel Nothman, Tara Murphy, and James R Curran. 2009. Analysing wikipedia and gold-standard corpora for ner training. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620. Association for Computational Linguistics.

Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, Nathan Schneider, and Timothy Baldwin. 2015. Big data small data, in domain out-of domain, known word unknown word: The impact of word representation on sequence labelling tasks. *arXiv preprint arXiv:1504.05319*.

Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R South, Danielle L Mowery, Gareth JF Jones, et al. 2013. Overview of the share/clef ehealth evaluation lab 2013. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 212–231. Springer.

Buzhou Tang, Yonghui Wu, Min Jiang, Joshua C Denny, and Hua Xu. 2013. Recognizing and encoding discorder concepts in clinical text using machine learning and vector space model. In *Workshop of ShARe/CLEF eHealth Evaluation Lab 2013*.

Buzhou Tang, Hongxin Cao, Xiaolong Wang, Qingcai Chen, and Hua Xu. 2014. Evaluating word representation features in biomedical named entity recognition tasks. *BioMed research international*, 2014.

Buzhou Tang, Yudong Feng, Xiaolong Wang, Yonghui Wu, Yaoyun Zhang, Min Jiang, Jingqi Wang, and Hua Xu. 2015. A comparison of conditional random fields and structured support vector machines for chemical entity recognition in biomedical literature. *Journal of cheminformatics*, 7(supplement 1).

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

E Voorhees and R Tong. 2011. Overview of the trec 2011 medical records track. In *Proceedings of TREC*, volume 4.

Dominic Widdows and Trevor Cohen. 2014. Reasoning with vectors: A continuous model for fast robust inference. *Logic Journal of IGPL*, pages 141–173.

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Using Entity Information from a Knowledge Base to Improve Relation Extraction

Lan Du[1]*, Anish Kumar[2], Mark Johnson[2] and Massimiliano Ciaramita[3]
[1]Faculty of Information Technology, Monash University Clayton, VIC 3800, Australia
[2]Department of Computing, Macquarie University Sydney, NSW 2109, Australia
[3]Google Research, Zurich, Switzerland

Abstract

Relation extraction is the task of extracting predicate-argument relationships between entities from natural language text. This paper investigates whether background information about entities available in knowledge bases such as FreeBase can be used to improve the accuracy of a state-of-the-art relation extraction system. We describe a simple and effective way of incorporating FreeBase's *notable types* into a state-of-the-art relation extraction system (Riedel et al., 2013). Experimental results show that our notable type-based system achieves an average 7.5% weighted MAP score improvement. To understand where the notable type information contributes the most, we perform a series of ablation experiments. Results show that the notable type information improves relation extraction more than NER labels alone across a wide range of entity types and relations.

1 Introduction

The goal of relation extraction is to extract relational information about entities from a large text collection. For example, given the text "Michael Bay, the director of Transformers, visited Paris yesterday," a relation extraction system might extract the relationship *film_director(Michael Bay, Transformers)*. These tuples can be then used to extend a knowledge base. With the increase in the amount of textual data available on the web, relation extraction has gained wide applications in information extraction from both general newswire texts and specialised document collections such as biomedical texts (Liu et al., 2007).

A typical relation extraction system functions as a pipeline, first performing named entity recognition (NER) and entity disambiguation to link the entity mentions found in sentences to their database entries (e.g., "Michael Bay" and "Transformers" would both be linked to their respective database ids). Then the context in which these entity mentions co-occur is used to predict the relationship between the entities. For example, the path in a syntactic parse between two mentions in a sentence can be used as a feature to predict the relation holding between the two entities. Continuing our example, the text pattern feature *X-the-director-of-Y* (or a corresponding parse subtree fragment) might be used to predict the database relation *film_director(X,Y)*. In such a pipeline architecture, information about the entities from the database is available and can be used to help determine the most appropriate relationship between the entities. The goal of this paper is to identify whether that information is useful in a relation extraction task, and study such information about the entities with a set of ablation experiments.

We hypothesise that information from database entries can play the role of background knowledge in human sentence comprehension. There is strong evidence that humans use world knowledge and contextual information in both syntactic and semantic interpretation (Spivey-Knowlton and Sedivy, 1995), so it is reasonable to expect a machine might benefit from it as well. Continuing with our example, if our database contained the information that one particular entity with the name Michael Bay had died a decade before the movie Transformers was released, then it might be reasonable to conclude that this particular individual was unlikely to have directed Transformers. Clearly, modelling all the ways in which such background information about entities might be used would be extremely complex. This paper explores a simple way of using some of the background information

*This work was partially done while Lan Du was at Macquarie University.

Lan Du, Anish Kumar, Mark Johnson and Massimiliano Ciaramita. 2015. Using Entity Information from a Knowledge Base to Improve Relation Extraction . In *Proceedings of Australasian Language Technology Association Workshop*, pages 31−38.

about entities available in FreeBase (Bollacker et al., 2008).

Here we focus on one particular kind of background information about entities — the information encoded in FreeBase's *notable types*. FreeBase's notable types are simple atomic labels given to entities that indicate what the entity is notable for, and so serve as a useful information source that should be relatively easy to exploit. For example, the search results for "Jim Jones" given by FreeBase contains several different entities. Although they all have the same name entity (NE) category PERSON, their notable types are different. The notable types for the top 4 "Jim Jones" results are *organization/organization_founder*, *music/composer*, *baseball/baseball_player* and *government/politician*. It is clear that the notable type information provides much finer-grained information about "Jim Jones" than just the NE category. It is reasonable to expect that notable types would be useful for relation extraction; e.g., the politician Jim Jones is likely to stand for election, while the baseball player is likely to be involved in sport activities.

We extend one state-of-the-art relation extraction system of Riedel et al. (2013) to exploit this notable type information. Our notable type extensions significantly improve the mean averaged precision (MAP) by 7.5% and the weighted MAP by 6% over a strong state-of-the-art baseline. With a set of ablation experiments we further evaluate how and where the notable type information contributes to relation extraction.The rest of this paper is structured as follows. The next section describes related work on relation extraction. Section 3 describes how a state-of-the-art relation extraction system can be extended to exploit the notable type information available in FreeBase. Section 4 specifies the inference procedures used to identify the values of the model parameters, while section 5 explains how we evaluate our models and presents a systematic experimental comparison of the models by ablating the notable type in different ways based on entities' NE categories. Section 6 concludes the paper and discusses future work.

2 Related work

Most approaches to relation extraction are either supervised or semi-supervised. Supervised approaches require a large set of manually annotated text as training data (Culotta and Sorensen, 2004), but creating these annotations is both expensive and error-prone. Semi-supervised approaches, by contrast, rely on correlations between relations and other large data sources.

In relation extraction, most semi-supervised approaches use *distant supervision*, which aligns facts from a large database, e.g., Freebase, to unlabelled text by assuming some systematic relationship between the documents and the database (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2010; Yao et al., 2010). Typically, we assume that (a) an entity linker can reliably identify entity mentions in the text and map them to the corresponding database entries, and (b) for all tuples of entities that appear in a relation in the database, if we observe that entity tuple co-occurring in a suitable linguistic construction (e.g., a sentence) then that construction expresses the database relationship about those entities. Previous work (Weston et al., 2013; Riedel et al., 2013; Bordes et al., 2013; Chang et al., 2014) has shown that models leveraging rich information from database often yield improved performance.

In this work we are particularly interested in exploring entity type information in relation extraction, as semantic relations often have selectional preference over entity types. Yao et al. (2010), Singh et al. (2013), Yao et al. (2013), Koch et al. (2014) and Chang et al. (2014) have shown that the use of type information, e.g., NE categories, significantly improves relation extraction. Our work here is similar except that we rely on Freebase's notable types, which provide much finer-grained information about entities. One of the challenges in relation extraction, particularly when attempting to extract a large number of relations, is to generalise appropriately over both entities and relations. Techniques for inducing *distributed vector-space representations* can learn embeddings of both entities and relations in a high-dimensional vector space, providing a natural notion of similarity (Socher et al., 2013) that can be exploited in the relation extraction task (Weston et al., 2013). Instead of treating notable types as features Ling and Weld (2012), here we learn distributed vector-space representations for notable types as well as entities, entity tuples and relations.

3 Relation extraction as matrix completion

Riedel et al. (2013) formulated the relation extraction task as a matrix completion problem. In this section we extend this formulation to exploit notable types in a simple and effective way. Specifically, we follow Riedel et al. (2013) in assuming that our data $\mathcal{O}$ consists of pairs $\langle r, t\rangle$, where $r \in \mathcal{R}$ is a relation and $t \in \mathcal{T}$ is a tuple of entities. The tuples are divided into training and test depending on which documents they are extracted from. In this paper, the tuples in $\mathcal{T}$ are always pairs of entities, but nothing depends on this. There are two kinds of relations in $\mathcal{R}$: syntactic patterns found in the document collection, and those appearing in the database (including target relations for extraction). For our notable type extension we assume we have a function n that maps an entity e to its FreeBase notable type $n(e)$.

For example, given text "Michael Bay, the director of Transformers, visited Paris yesterday" we extract the pair $\langle r, t\rangle$ where $t = \langle$*Michael Bay, Transformers*$\rangle$ and $r =$ *X-the-director-of-Y* (actually, the path in a dependency parse between the named entities). From FreeBase we extract the pair $\langle r', t\rangle$ where $r' =$ *film/director*. FreeBase also tells us that $n($*Michael Bay*$) =$ *Person* and $n($*Transformers*$) =$ *Film*. Our goal is to learn a matrix $\boldsymbol{\Theta}$ whose rows are indexed by entity tuples in $\mathcal{T}$ and whose columns are indexed by relations in $\mathcal{R}$. The entry $\theta_{t,r}$ is the *log odds of relation* $r \in \mathcal{R}$ *holding of tuple* $t \in \mathcal{T}$, or, equivalently, the probability that relation r holds of tuple t is given by $\sigma(\theta_{t,r})$, where σ is the logistic function: $\sigma(x) = (1 + e^{-x})^{-1}$.

Riedel et al. (2013) assume that $\boldsymbol{\Theta}$ is the sum of three submodels: $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{\mathrm{N}} + \boldsymbol{\Theta}^{\mathrm{F}} + \boldsymbol{\Theta}^{\mathrm{E}}$, where $\boldsymbol{\Theta}^{\mathrm{N}}$ is the neighbourhood model, $\boldsymbol{\Theta}^{\mathrm{F}}$ is the latent feature model and $\boldsymbol{\Theta}^{\mathrm{E}}$ is the entity model (these will be defined below). Here we extend these submodels using FreeBase's notable types.

3.1 A notable type extension to the neighbourhood model

The neighbourhood model $\boldsymbol{\Theta}^{\mathrm{N}}$ captures dependencies between the syntactic relations extracted from the text documents and the database relations extracted from FreeBase. This is given by:

$$\theta^{\mathrm{N}}_{r,t} \;=\; \sum_{\langle r',t\rangle \in \mathcal{O}\setminus\{\langle r,t\rangle\}} w_{r,r'},$$

where $\mathcal{O}$ is the set of relation/tuple pairs in the data and $\mathcal{O} \setminus \{\langle r, t\rangle\}$ is $\mathcal{O}$ with the tuple $\langle r, t\rangle$ removed. $\boldsymbol{w}$ is a matrix of parameters, where $w_{r,r'}$ is a real-valued weight with which relation r' "primes" relation r that will be learnt from the training data. The neighbourhood model can be regarded as predicting an entry $\theta_{r,t}$ by using entries along the same row. It functions as a logistic regression classifier predicting the log odds of a FreeBase relation r applying to the entity tuple t using as features the syntactic relations r' that hold of t.

Our notable type extension to the neighbourhood model enriches the syntactic patterns in the training data $\mathcal{O}$ with notable type information. For example, if there is a syntactic pattern for *X-director-of-Y* in our training data (say, as part of the tuple $\langle$*X-director-of-Y*, $\langle$*Michael Bay, Transformers*$\rangle\rangle$), then we add a new syntactic pattern $\langle$*Person(X)-director-of-Film(Y)*$\rangle$, where *Person* and *Film* are notable types and add the tuple $\langle$*Person(X)-director-of-Film(Y)*, $\langle$*Michael Bay, Transformers*$\rangle\rangle$ to our data $\mathcal{O}$. Each new relation corresponds to a new column in our matrix completion formulation. More precisely, the new relations are members of the set $\mathcal{N} = \{\langle r, n(t)\rangle : \langle r, t\rangle \in \mathcal{O}\}$, where $n(t)$ is the tuple of notable types corresponding to the entity tuple t. For example, if $t = \langle$*Michael Bay, Transformers*$\rangle$ then $n(t) = \langle$*Person, Film*$\rangle$. Then the notable type extension of the neighbourhood model is:

$$\theta^{\mathrm{N}'}_{r,t} \;=\; \sum_{\langle r',t\rangle \in \mathcal{O}\setminus\{\langle r,t\rangle\}} w_{r,r'} + w'_{r,\langle r',n(t)\rangle}$$

where $\boldsymbol{w}'$ is a matrix of weights relating the relations $\mathcal{N}$ to the target FreeBase relation r.

3.2 A notable type extension to the latent feature model

The latent feature model generalises over relations and entity tuples by associating each of them with a 100-dimensional real-valued vector. Intuitively, these vectors organise the relations and entity tuples into clusters where conceptually similar relations and entity tuples are "close," while those that are dissimilar are far apart. In more detail, each relation $r \in \mathcal{R}$ is associated with a latent feature vector $\boldsymbol{a}_r$ of size $K = 100$. Similarly, each entity tuple $t \in \mathcal{T}$ is also associated with a latent feature vector $\boldsymbol{v}_t$ of size K as well. Then the latent feature score for an entity tuple t and relation r is just the dot product of the corresponding relation and entity tuple vectors, i.e.: $\theta^{\mathrm{F}}_{r,t} = \boldsymbol{a}_r \cdot \boldsymbol{v}_t$.

We extend the latent feature model by associating a new latent feature vector with each notable type sequence observed in the training data, and use this vector to enrich the vector-space representations of the entity tuples. Specifically, let $\mathcal{T}' = \{n(t) : t \in \mathcal{T}\}$ be the set of notable type tuples for all of the tuples in $\mathcal{T}$, where $n(t)$ is the tuple of notable types corresponding to the tuple of entities t as before. We associate each tuple of notable types $t' \in \mathcal{T}'$ with a latent feature vector $v'_{t'}$ of dimensionality K. Then we define the notable type extension to the latent feature model as:

$$\theta^{\mathrm{F}'}_{r,t} \;=\; \boldsymbol{a}_r \cdot (\boldsymbol{v}_t + \boldsymbol{v}'_{n(t)}) \,.$$

This can be understood as associating each entity tuple $t \in \mathcal{T}$ with a pair of latent feature vectors $\boldsymbol{v}_t$ and $\boldsymbol{v}_{n(t)}$. The vector $\boldsymbol{v}_{n(t)}$ is based on the notable types of the entities, so it can capture generalisations over those notable types. The L_2 regularisation employed during inference prefers latent feature vectors in which $\boldsymbol{v}_t$ and $\boldsymbol{v}'_{n(t)}$ are small, thus encouraging generalisations which can be stated in terms of notable types to be captured by $\boldsymbol{v}'_{n(t)}$.

3.3 A notable type extension of the entity model

The entity model represents an entity e with a K-dimensional ($K = 100$) feature vector u_e. Similarly, the ith argument position of a relation r is also represented by a K-dimensional feature vector $d_{r,i}$. The entity model associates a score $\theta^{\mathrm{E}}_{r,t}$ with a relation $r \in \mathcal{R}$ and entity tuple $t \in \mathcal{T}$ as follows: $\theta^{\mathrm{E}}_{r,t} = \sum_{i=1}^{|t|} \boldsymbol{d}_{r,i} \cdot \boldsymbol{u}_{t_i}$, where $|t|$ is the arity of (i.e., number of elements in the entity tuple t), t_i is the ith entity in the entity tuple t, and $\boldsymbol{d}_{r,i}$ and $\boldsymbol{u}_{t_i}$ are K-dimensional vectors associated with the ith argument slot of relation r and the entity t_i respectively. The intuition is that the latent feature vectors of co-occurring entities and argument slots should be close to each other in the K-dimensional latent feature space, while entities and argument slots that do not co-occur should be far apart.

Our notable type extension of the entity model is similar to our notable type extension of the latent feature model. We associate each notable type m with a K-dimensional feature vector $\boldsymbol{u}'_m$, and use those vectors to define the entity model score. Specifically, the entity model score is defined as:

$$\theta^{\mathrm{E}'}_{r,t} \;=\; \sum_{i=1}^{|t|} \boldsymbol{d}_{r,i} \cdot \left(\boldsymbol{u}_{t_i} + \boldsymbol{u}'_{n(t_i)}\right) \,,$$

where $n(e)$ is the notable type for entity e and $|t|$ is the length of tuple t. The L2 regularisation again should encourage generalisations that can be expressed in terms of notable types to be encoded in the $\boldsymbol{u}'_{n(t_i)}$ latent feature vectors.

4 Inference for model parameters

The goal of inference is to identify the values of the model's parameters, i.e., $\boldsymbol{w}, \boldsymbol{a}, \boldsymbol{v}, \boldsymbol{d}$ and $\boldsymbol{u}$ in the case of the Riedel et al model, and these plus $\boldsymbol{w}', \boldsymbol{v}'$ and $\boldsymbol{u}'$ in the case of the notable type extensions. The inference procedure is inspired by Bayesian Personalised Ranking (Rendle et al., 2009). Specifically, while the true value of $\theta_{r,t}$ is unknown, it's reasonable to assume that if $\langle r, t^+\rangle \in \mathcal{O}$ (i.e., is observed in the training data) then $\theta_{r,t^+} > \theta_{r,t^-}$ for all $\langle r, t^-\rangle \notin \mathcal{O}$ (i.e., not observed in the training data). Thus the training objective is to maximise

$$\ell \;=\; \sum_{\langle r,t^+\rangle \in \mathcal{O}} \; \sum_{\langle r,t^-\rangle \notin \mathcal{O}} \ell_{\langle r,t^+\rangle,\langle r,t^-\rangle}$$

where: $\ell_{\langle r,t^+\rangle,\langle r,t^-\rangle} = \log \sigma(\theta_{r,t^+} - \theta_{r,t^-})$, and $\theta_{r,t} = \theta^{\mathrm{N}}_{r,t} + \theta^{\mathrm{F}}_{r,t} + \theta^{\mathrm{E}}_{r,t}$ or $\theta_{r,t} = \theta^{\mathrm{N}'}_{r,t} + \theta^{\mathrm{F}'}_{r,t} + \theta^{\mathrm{E}'}_{r,t}$, depending on whether the submodels with notable type extensions are used. The objective function ℓ is then maximised by using stochastic gradient ascent. The stochastic gradient procedure sweeps through the training data, and, for each observed tuple $\langle r, t^+\rangle \in \mathcal{O}$, samples a negative evidence tuple $\langle r, t^-\rangle \notin \mathcal{O}$ not in the training data, adjusting weights to prefer the observed tuple.

In our experiments below we ran stochastic gradient ascent with a step size of 0.05 and an L2 regulariser constant of 0.1 for the neighbourhood model and 0.01 for the latent feature and entity models (we used the same regulariser constants for models both with and without the notable type extensions). We ran 2,000 sweeps of stochastic gradient ascent.

5 Experimental evaluation

We used a set of controlled experiments to see to what extent the notable type information improves the state-of-the-art relation extraction system. We used the New York Times corpus (Sandhaus, 2008) in our experiments, assigning articles from the year 2000 as the training corpus and the articles from 1990 to 1999 for testing. The entity tuples $\mathcal{T}$ were extracted from the New York Times corpus (tuples that did not appear at least 10 times and also appear in one of the FreeBase relations were discarded). The relations $\mathcal{R}$ are either syntactic patterns found in the New York Times corpus, FreeBase relations, or (in our extension) notable types extracted from FreeBase. Our evaluation focuses on 19 FreeBase relations, as in Riedel et al. (2013).

5.1 Notable type identification

Our extension requires a FreeBase notable type for every entity mention, which in turn requires a Freebase entity id because a notable type is a property associated with entities in FreeBase. We found the entity id for each named entity as follows. We used the FreeBase API to search for the notable type for each named entity mentioned in the training or test data. In cases where several entities were returned, we used the notable type of the first entity returned by the API. For example, the FreeBase API returns two entities for the string "Canada:" a country and a wine (in that order), so we use the notable type "country" for "Canada" in our experiments. This heuristic is similar to the method of choosing the most likely entity id for a string, which provides a competitive baseline for entity linking (Hoffart et al., 2011).

5.2 Evaluation procedure

After the training procedure is complete and we have estimates for the model's parameters, we can use these to compute estimates for the log odds $\theta_{r,t}$ for the test data. These values quantify how likely it is that the FreeBase relation r holds of an entity tuple t from the test set, according to the trained model.

In evaluation we follow Riedel et al. (2013) and treat each of the 19 relations r as a query, and evaluate the ranking of the entity tuples t returned according to $\theta_{r,t}$. For each relation r we pool the highest-ranked 100 tuples produced by each of the models and manually evaluate their accuracy (e.g., by inspecting the original document if necessary). This gives a set of results that can be used to calculate a precision-recall curve. Averaged precision (AP) is a measure of the area under that curve (higher is better), and mean average precision (MAP) is average precision averaged over all of the relations we evaluate on. Weighted MAP is a version of MAP that weights each relation by the true number of entity tuples for that relation (so more frequent relations count more).

An unusual property of this evaluation is that increasing the number of models being evaluated generally decreases their MAP scores: as we evaluate more models, the pool of "true" entity tuples for each relation grows in size and diversity (recall that this pool is manually constructed by manually annotating the highest-scoring tuples returned by each model). Thus in general the recall scores of the existing models are lowered as the number of models increases.

Relation	#	NF	NF^T	NFE	NFE^T
person/company	131	0.83	0.89	0.83	0.86
location/containedby	88	0.68	0.69	0.68	0.69
person/nationality	51	0.11	0.55	0.15	0.45
author/works_written	38	0.51	0.53	0.57	0.53
person/parents	34	0.14	0.31	0.11	0.28
parent/child	31	0.48	0.58	0.49	0.58
person/place_of_birth	30	0.51	0.48	0.56	0.57
person/place_of_death	22	0.75	0.77	0.75	0.77
neighbourhood/neighbourhood_of	17	0.48	0.55	0.52	0.54
broadcast/area_served	8	0.21	0.41	0.26	0.30
company/founders	7	0.46	0.27	0.40	0.28
team_owner/teams_owned	6	0.21	0.25	0.25	0.25
team/arena_stadium	5	0.06	0.07	0.06	0.09
film/directed_by	5	0.21	0.25	0.24	0.35
person/religion	5	0.20	0.28	0.21	0.23
composer/compositions	4	0.42	0.44	0.06	0.42
sports_team/league	4	0.70	0.62	0.63	0.64
film/produced_by	3	0.17	0.30	0.12	0.26
structure/architect	2	*1.00*	*1.00*	*1.00*	*1.00*
MAP		0.43	**0.49**	0.42	0.48
Weighted MAP		0.55	**0.64**	0.56	0.62

Table 1: Averaged precision and mean average precision results. The rows correspond to FreeBase relations, and the columns indicate the combination of sub-models (N = neighbourhood model, F = latent feature model, E = entity model). The superscript "T" indicates the combined models that incorporate the notable type extensions, and the # column gives the number of true facts.

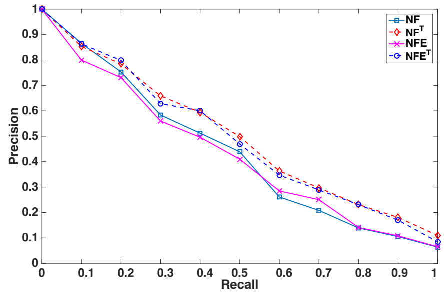


Figure 1: Averaged 11-point precision-recall curve for the four models shown in Table 1.

5.3 Experiments with Notable Types

We found we obtained best performance from the model that incorporates all submodels (which we call NFE^T) and from the model that only incorporates the Neighbourhood and Latent Feature submodels (which we call NF^T), so we concentrate on them here. Table 1 presents the MAP and weighted MAP scores for these models on the 19 FreeBase relations in the testing set.

The MAP scores are 6% higher for both NF^T and NFE^T, and the weighted MAP scores are 9% and 6% higher for NF^T and NFE^T respectively.

Relation	#	NE	NFE^T	NE+P	NE+L	NE+O	NE+M
person/place_of_birth	30	0.52	**0.57**	0.54	0.50	0.50	0.54
author/works_written	38	0.57	0.53	**0.61**	0.56	0.57	0.49
team/arena_stadium	5	0.08	0.09	**0.10**	0.09	0.07	0.09
composer/compositions	4	0.35	0.42	**0.51**	0.37	0.35	0.45
person/company	131	0.81	**0.86**	0.84	0.82	0.83	**0.86**
film/directed_by	5	0.30	0.35	**0.41**	0.27	0.27	**0.41**
neighbourhood/neighbourhood_of	17	0.59	0.54	0.59	0.49	0.59	**0.62**
film/produced_by	3	0.20	0.26	0.29	0.18	0.19	**0.40**
person/religion	5	0.22	0.23	0.21	0.22	0.28	**0.53**
location/containedby	88	0.66	0.69	0.68	0.64	0.64	**0.70**
sports_team/league	4	0.53	0.64	0.54	0.52	**0.75**	**0.75**
person/parents	34	0.33	0.28	0.30	0.32	**0.35**	0.34
parent/child	31	0.55	0.58	0.56	0.55	**0.59**	0.56
person/place_of_death	22	0.71	0.77	0.74	0.74	**0.78**	0.72
company/founders	7	0.22	0.28	0.28	0.21	**0.29**	0.22
team_owner/teams_owned	6	0.34	0.25	0.27	0.34	**0.36**	0.35
person/nationality	51	0.19	0.45	0.23	**0.50**	0.20	0.21
broadcast/area_served	8	0.32	0.30	0.33	**0.38**	0.31	0.29
structure/architect	2	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*
MAP		0.45	0.48	0.48	0.46	0.47	**0.50**
Weighted MAP		0.57	**0.62**	0.59	0.60	0.58	0.60

Table 2: Results of ablation experiments on the NFE^T model. The columns correspond to experiments, and the column labels are explained in Table 3.

A sign test shows that the difference between the models with notable types and those without the notable types is statistically significant ($p < 0.05$). Clearly, the notable type extensions significantly improve the accuracy of the existing relation extraction models. Figure 1 shows an averaged 11-point precision-recall curve for these four models. This makes clear that across the range of precision-recall trade-offs, the models with notable types offer the best performance.

5.4 Ablation Experiments

We performed a set of ablation experiments to determine exactly how and where the notable type information improves relation extraction. In these experiments entities are divided into 4 "named entity" (NE) classes, and we examine the effect of just providing notable type information for the entities of a single NE class. The 4 NE classes we used were PERSON, LOCATION, ORGANISATION, and MISC (miscellaneous). We classified all entities into these four categories using their FreeBase types, which provide a more coarse-grained classification than notable types. For example, if an entity has a FreeBase "*people/person*" type, then we assigned it to the NE class PERSON; if an entity has a "*location/location*" type, then its NE class is LOCATION; and if an entity has a "*organisation/organisation*" type, then its NE class is ORGANISATION. All entities not classified as PERSON, LOCATION, or ORGANISATION were labelled MISC.

We ran a set of ablation experiments as follows. For each NE class c in turn, we replaced the notable type information for entities not classified as c with their NE class. For example, when c = PERSON, only entities with the NE label PERSON had notable type information, and the notable types of all other entities was replaced with their NE labels. Table 3 lists the different ablation experiments. The ablation experiments are designed to study which NE classes the notable types help most on. The results are reported in Table 2. The results clearly indicate that different relations benefit from the different kinds of notable type information about entities.

Ablation setting	Description
NE	All entities are labelled with their NE class instead of their notable type.
NE+P	Only PERSON entities have notable type information; the notable type of other entities is replaced with their NE class.
NE+L	Only LOCATION entities have notable type information; the notable type of other entities is replaced with their NE class.
NE+O	Only ORGANISATION entities have notable type information; the notable type of other entities is replaced with their NE class.
NE+M	Only MISC entities have notable type information; the notable type of other entities is replaced with their NE class.

Table 3: Descriptions of the ablation experiments in Table 2.

Column "NE+P" shows that relations such as "*author/works_written*", "*composer/compositions*" and "*film/directed_by*" benefit the most from notable type information about PERSONs. We noticed that there are about 43K entities classified as PERSON, which includes 8,888 book authors, 802 music composers, 1212 film directors, etc. These entities have 214 distinct notable types. Our results show that it is helpful to distinguish the PERSON entities with their notable types for relations involving professions. For example, not all people are authors, so knowing that a person is an author increases the accuracy of extracting "*author/works_written*". Similarly, Column "NE+L" shows that "*person/nationality*" and "*broadcast/area_served*" gain the most from the notable type information about locations. There are about 8.5K entities classified as LOCATION, which includes 4807 city towns, 301 countries, and so on. There are 170 distinct notable types for LOCATION entities.

Column "NE+O" shows that the notable type information about ORGANISATION entities improves the accuracy of extracting relations involving organisations. Indeed, there are more than

3K business companies and 200 football teams. Notable type information about organisations improves extraction of the "*parent/child*" relation because this relation involves entities such as companies. For example, in our corpus the sentence "CNN, a unit of the Turner Broadcasting, says that 7000 schools have signed up for The Newsroom" expresses the *parent/child(Turner Broadcasting, CNN)* relation.

The ablation results in Column "NE+M" show that information about MISC entities is most useful of all, as this ablation experiment yielded the highest overall MAP score. There are about 13.5K entities labelled MISC. The most frequent notable types for entities in the MISC NE class are "*film/film*" and "*book/book*". Therefore it is reasonable that notable type information for MISC entities would improve AP scores for relations such as "*film/directed_by*" and "*person/religion*". For example, "George Bush reached a turning point in his life and became a born-again Christian" is an example of the "*person/religion*" relation, and it's clear that it is useful to know that "born-again Christian" belongs to the religion notable type. The "sports_team/league" relation is interesting because it performs best with notable type information for entities in the ORGANISATION or MISC NE classes. It turns out that roughly half the sports teams are classified as ORGANISATIONs and half are classified as MISC. The sports teams that are classified as MISC are missing the "organisation/organisation" type in their FreeBase entries, otherwise they would be classified as ORGANISATIONs.

In summary, the ablation results show that the contribution of notable type information depends on the relation being extracted. The result demonstrates that relations involving organisations benefits from the notable type information about these organisations. It also demonstrates that certain relations benefit more from notable type information than others. Further research is needed understand some of the ablation experiment results (e.g., why does person/place of death perform best with notable type information about ORGANISATIONs?)

6 Conclusion and future work

In this paper we investigated the hypothesis that background information about entities present in a large database such as FreeBase can be useful for relation extraction. We modified a state-of-the-art relation extraction system (Riedel et al., 2013) by extending each of its submodels to exploit the "notable type" information about entities available in FreeBase. We demonstrated that these extensions improve the MAP score by 6% and the weighted MAP score by 7.5%, which is a significant improvement over a strong baseline. Our ablation experiments showed that the notable type information improves relation extraction more than NER tags across a wide range of entity types and relations.

In future work we would like to develop methods for exploiting other information available in FreeBase to improve a broad range of natural language processing and information extraction tasks. We would like to explore ways of exploiting entity information beyond (distant) supervision approaches, for example, in the direction of OpenIE (Wu and Weld, 2010; Fader et al., 2011; Mausam et al., 2012). The temporal information in a large database like FreeBase might be especially useful for named entity linking and relation extraction: e.g., someone that has died is less likely to release a hit single. In summary, we believe that there are a large number of ways in which the rich and diverse information present in FreeBase might be leveraged to improve natural language processing and information retrieval, and exploiting notable types is just one of many possible approaches.

Acknowledgments

This research was supported by a Google award through the Natural Language Understanding Focused Program, and under the Australian Research Council's *Discovery Projects* funding scheme (project numbers DP110102506 and DP110102593).

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 423–429.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 1535–1545.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)* , pages 782–792.

Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1891–1901.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *AAAI*.

Yudong Liu, Zhongmin Shi, and Anoop Sarkar. 2007. Exploiting rich syntactic information for relation extraction from biomedical articles. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-Short'07, pages 97–100.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, ECML PKDD'10, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Evan Sandhaus. 2008. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, pages 1–6.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems (NIPS 26)*, pages 926–934.

Michael Spivey-Knowlton and Julie Sedivy. 1995. Resolving attachment ambiguities with multiple constraints. *Cognition*, 55:227–267.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371.

Fei Wu and Daniel S. Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL'10, pages 118–127.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, pages 79–84.

Likelihood Ratio-based Forensic Voice Comparison on L2 speakers: A Case of Hong Kong native male production of English vowels

Daniel Frost[1], **Shunichi Ishihara**[2,3]
[1]School of Language, Literature and Linguistics, The Australian National University, Australia
[2]Department of Linguistics, The Australian National University, Australia
[3]The Internet Commerce Security Laboratory, Federation University, Australia
drbfrost@live.com.au, shunichi.ishihara@anu.edu.au

Abstract

This study is a pilot research that explores the effectiveness of a likelihood ratio (LR)-based forensic voice comparison (FVC) system built on non-native speech production. More specifically, it looks at native Hong Kong Cantonese-speaking male productions of English vowels, and the extent to which FVC can work on these speakers. 15 speakers participated in the research, involving two non-contemporaneous recording sessions with six predetermined target words – “hello”, “bye”, “left”, “right”, “yes”, and “no”. Formant frequency values were measured from the trajectories of the vowels and surrounding segments. These trajectories were modelled using discrete cosine transforms for each formant (F1, F2 and F3), and the coefficient values were used as feature vectors in the LR calculations. LRs were calculated using the multivariate-kernel-density method. The results are reported along two metrics of performance, namely the log-likelihood-ratio cost and 95% credible intervals. The six best-performing word-specific outputs are presented and compared. We find that FVC can be built using L2 speech production, and the results are comparable to similar systems built on native speech.

1 Introduction

1.1 Forensic voice comparison and the likelihood-ratio framework

Forensic voice comparison (FVC) is the forensic science of comparing voices. It is most often used in legal contexts where the origin of voice samples is being debated. Typically, an FVC analysis involves the comparison of voice recordings of known origin (e.g. the suspect’s speech samples) with other voice recordings of disputed origin (e.g. the offender’s speech samples) (Rose, 2004). The FVC expert will apply statistical techniques on data extracted from speech sample evidence with the ultimate aim of assisting the trier of fact (e.g. judge(s)/jury) with their final decision. The trier of fact is faced with the task of making this decision by analysing the numerous probabilistic forms of evidence offered to them over the course of the trial. In fact, this decision is in itself a probabilistic statement, known as the posterior odds, and can be expressed mathematically as 1).

$$\frac{p(H|E)}{p(\bar{H}|E)} \qquad (1)$$

In 1), $p(H|E)$ represents the probability of one hypothesis (e.g. the prosecution hypothesis – the suspect is guilty), given the various forms of evidence (e.g. DNA, fingerprint, voice, witness accounts etc.), and $p(\bar{H}|E)$ represents the probability of the alternative hypothesis (e.g. the defence hypothesis – the suspect is not guilty), given the evidence. In the context of FVC, 1) becomes:

$$\frac{p(H_{SS}|E)}{p(H_{DS}|E)} \qquad (2)$$

In 2), H_{SS} represents the same-speaker hypothesis, and H_{DS} represents the different-speaker hypothesis. Before the trier of fact is able to make their decision of guilt or innocence, there may be more evidence that needs to be taken into account (e.g. DNA, fingerprint, witness etc.), and the FVC expert does not have access to this evidence (Rose, 2002, p. 57). If the FVC expert were to provide the

Daniel Frost and Shunichi Ishihara. 2015. Likelihood Ratio-based Forensic Voice Comparison on L2 speakers: A Case of Hong Kong native male production of English vowels . In *Proceedings of Australasian Language Technology Association Workshop*, pages 39−47.

trier of fact with this strength-of-hypotheses statement, they would in effect be making a statement about the suspect's guilt or innocence, which is usurping the role of the trier of fact (Aitken, 1995, p. 4; Evett, 1998; Morrison, 2009a, p. 300). This issue is resolved through the application of Bayes' Theorem, given in 3).

$$\underbrace{\frac{p(H_{SS}|E)}{p(H_{DS}\,|E)}}_{posterior\ odds} = \underbrace{\frac{p(E|H_{SS})}{p(E|H_{DS}\,)}}_{likelihood\,ratio} * \underbrace{\frac{p(H_{SS})}{p(H_{DS}\,)}}_{prior\ odds} \qquad (3)$$

By using the LR framework, the FVC expert (and the DNA expert, the fingerprint expert, etc.) is able to make an objective statement regarding the strength of the evidence, and in doing so, does not usurp the role of the trier of fact.

Put simply, the LR is the probability that some evidence would occur if an assertion is true, relative to the probability that the evidence would occur if the assertion is not true (Robertson & Vignaux, 1995, p. 17). The FVC-based LR above can be interpreted as the probability p of observing some evidence E (in FVC, this is the difference between the suspect and offender speech samples) if the same-speaker hypothesis H_{SS} is true, relative to the probability p of observing the same evidence E if the different-speaker hypothesis H_{DS} is true. For example, a calculated LR of 100 would be interpreted as follows: "the evidence is 100 times more likely to arise if the speech samples are of the same speaker, than it is if the speech samples are of different speakers". To emphasise, this is not the same as saying: "it is 100 times more likely that the speech samples are of the same speaker than of different speakers".

The process essentially involves calculating the similarity of two samples as well as the typicality of the two samples against a relevant background population. The similarity and typicality are the numerator and denominator of the LR respectively.

1.2 Non-native speakers (L2 speakers)

Since the National Institute of Standards and Technology (NIST) speaker recognition evaluations (SRE)[1] started including non-native speaker data (mostly English), a series of experiments have been carried out using L2 samples in non-forensic contexts (Durou, 1999; Kajarekar et al., 2009; Scheffer et al., 2011). However, until now, FVC research has been exclusively based on native (henceforth L1) speech production. However, crimes are obviously committed by L1 speakers and L2 speakers alike. There are therefore important practical applications to be developed from L2-based FVC research. To the best of our knowledge, this study is the first LR-based study exploring the effectiveness of an FVC system built on L2 speakers. While there have been studies that make considerations that could potentially apply to L2-based FVC, such as the selection of relevant reference samples (Morrison et al., 2012), there has not been an explicit attempt to build such a system.

The participants in this study spoke English had reasonably strong HK Cantonese "accents". Furthermore, they exhibited many tendencies of L2 speakers; stuttering, pausing to recall lexical items, using only a few set grammar patterns etc. However, we do not know how the phonetic characteristics of L2 speech affect between-speaker and within-speaker variations. One possibility is that L2 accents are not "hardwired" and therefore more fluid, potentially resulting in higher within-speaker variation; a hindrance for FVC.

1.3 Research question

Having briefly outlined the key concepts of the research, the research question is:

Can FVC work on non-native speech?

As the research question suggests, this study is exploratory in nature. We maintained tight control over many variables in order to eliminate some complexities that might arise in deeper research, in order to produce a baseline for future research. The reader should note that the aim is not to find the most effective method for L2-based FVC.

2 Research Design

Speech data were collected from 15 male speakers of Hong Kong (henceforth HK) Cantonese. We used a map task to elicit the voice samples. A map task is a simple speaking task in which the participant is provided a basic map, and the interviewer conducts a mock scenario asking for simple directions to certain places, or asks about general details of the map. The map task, conducted entirely in English, allows an interviewer to elicit large quantities of a set of words without reverting to a less natural word-list method.

[1] http://www.itl.nist.gov/iad/mig/tests/spk/

All speakers were 1) male; 2) over 18 years old; 3) HK natives; 4) identify as native speakers of HK Cantonese; and 5) completed their compulsory schooling in HK. Speakers were between 18 and 24 years of age (except one 42-year-old) and attended two non-contemporaneous recording sessions at least seven days apart (mean=12.86 days excluding an outlier of 80 days). The authors acknowledge that the number of speakers in the database is very small, though real FVC casework often involves analysis of limited amounts of data.

When performing word-specific FVC research, it is most suitable to work with common words in the English vernacular, keeping the practicalities of real casework in mind. The words given in Table 1 were chosen as the target words for both their phonetic properties and practical application. We decided to use 5 random tokens of each word to build the FVC system.

Word	**GAE broad transcription**	**HKE broad transcription**
hello	həl**əʉ**	hal**əʊ**
bye	b**ɑe**	b**aɪ**
left	l**e**ft	l**ɛ**ft
right	r**ɑe**t	r**aɪ**t
yes	**je**s	**jɛ**s
no	n**əʉ**	n**əʊ**

Table 1: Target words and broad transcriptions in GAE (General Australian English) (Harrington et al., 1997) and HKE (Hong Kong English) broad transcriptions. Target segments are in bold. Note that these transcriptions are merely representative of typical phoneme realisation.

The words in Table 1 are common English words and cover both monophthong vowel productions (stable single syllable peak with one articulatory target; "left", "yes") and diphthong vowel productions (dynamic single syllable peak with two distinct articulatory targets; "hello", "bye", "right", "no") (Cox, 2012, p. 29; Ladefoged & Disner, 2012, pp. 54-55). Diphthongs are commonly used in FVC research because they often have low within-speaker variation and high between-speaker variation. This is because a diphthong, unlike a monophthong, involves substantial movement of the formant trajectories, allowing more room for individualising information (Li & Rose, 2012, p. 202).

In our case, however, we have avoided labelling the vowels as “monophthong” or “diphthong”, because the data were extracted in a manner that captured both the formant trajectory of the vowel and the surrounding consonants and transitions where applicable. We are therefore dealing with differing levels of dynamism. Under this approach, “bye” and “right” are classed as being the most dynamic, and the least dynamic are “left”, and surprisingly, “hello”, in some speakers’ cases.

Each recording session was conducted in a soundproof recording studio using professional equipment. The recordings were made using the Audacity[2] software, preset for a 32 bit recording on a mono track at a 44.1 kHz sampling rate. They were later downsampled to 16 kHz.

The EMU Speech Database System[3] was used to analyse and annotate the recorded samples. The "forest" analysis application was used with the following settings: 3 formants to be defined (F1, F2, F3), Hamming window function with window size set to 25ms and window shift set to 5ms. The “forest” analysis performed very well in general.

2.1 Parametrisation

In order to build our FVC system, our formant trajectory portions needed to be modelled. We used a parametric curve fitting procedure that uses *discrete cosine transforms* (DCTs). The DCT method involves an estimation of a complex curve – the formant trajectories – by adding simple cosine functions together (Morrison, 2009b, p. 2389; Rose, 2013). These simple cosine functions are defined in terms of their coefficient values, which specify their amplitudes. The DCT coefficient values – from models of F1, F2, and F3 trajectories – were used as the feature vectors in the LR calculations. The durations of the trajectories were equalised because it has been shown to work well in FVC (Morrison, 2008, 2009b; Morrison & Kinoshita, 2008).

In this study, we use the term “output” to refer to the statistical and graphical result of a certain set of combinations of DCT coefficients and formants.

Figure 1 shows the modelled DCT curves (dotted lines) alongside the complex formant trajectories (solid lines) for all “bye” tokens. It is evident that higher degree DCT curves better approximate the complex formant trajectories.

[2] http://audacity.sourceforge.net/

[3] http://emu.sourceforge.net/

Table 2 shows the possible combinations of the parameters. Note that each output kept the DCT coefficient number constant across all formants in combination.

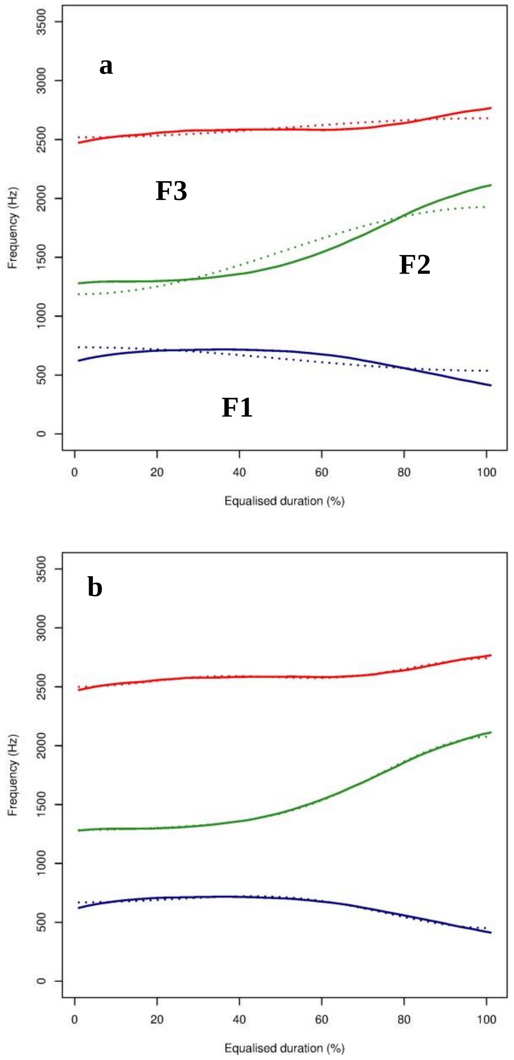


Figure 1: Solid lines represent the mean complex formant trajectories for all "bye" tokens in the dataset. The dotted lines represent 2nd degree (a) and 4th degree (b) DCT-modelled curves. X-axis = Equalised duration and Y-axis = Frequency in Hz.

3 Testing

In order to assess the performance of an FVC system, two types of comparisons, namely same-speaker (SS) and different-speaker (DS) comparisons, are necessary. In SS comparisons, two speech samples produced by the same individual are compared and evaluated with the derived LR. Given the same origin, it is expected that the derived LR is higher than 1. In DS comparisons, they are expected to receive an LR lower than 1. In total, there were 15 SS comparisons and 210 DS comparisons[4] for each target word.

Formant combination	DCT coefficients
f12, f23, f123	2, 3, 4, 5, 6, 7, 8, 9

Table 2: The multiple-formant output combinations ($6 \times 3 \times 8 = 144$ total combinations). Note that, for example, f12 represents results involving F1 and F2; f23 represents results involving F2 and F3, etc.

3.1 Multivariate-kernel-density procedure

One of the advantages of the LR framework is the ability to combine different pieces of evidence. If multiple LR values are obtained from different pieces of evidence (e.g. fingerprint, voice, DNA etc.), then these values may simply be multiplied together (added together in the logarithmic domain) to produce one LR value. This simple procedure, however, works under the assumption that the pieces of evidence are not correlated.

As explained in §2.1, DCT coefficients from models of F1, F2, and F3 trajectories were used as the feature vectors in the LR calculations. An issue here is the potential correlation between formants. The issue of correlated variables was addressed by Aitken & Lucy (2004) with their multivariate kernel density likelihood ratio (henceforth MVKD) formulae. By using a cross-validated MVKD procedure, we were able to obtain a single LR from multiple correlated features while taking the correlations into account (the statistical information for typicality is repeatedly recalculated from all samples except those speakers in comparison). The cross-validated MVKD approach has been used in many FVC studies (Ishihara & Kinoshita, 2008; Morrison, 2009b; Morrison & Kinoshita, 2008; Rose, 2013).

[4] For DS comparisons, two independent different DS comparisons are possible (e.g. (S)peaker1(R)ecording1 vs. S2R1 and S1R2 vs. S2R2) for each pair of different speakers (e.g. S1 vs. S2).

3.2 Logistic-regression calibration

When building an FVC system, raw output values may need to be calibrated before they are interpretable. The outputs of the MVKD calculations in §3.1 actually result in *scores*. Scores are logLRs in that their values indicate degrees of similarity between two speech samples having taken into account their typicality against a background populapopulation (Morrison, 2013, p. 2). Logistic-regression calibration (Brümmer & du Preez, 2006) is a method which converts these output scores to interpretable logLRs by performing a linear shift (in the logarithmic scale) on the scores relative to a decision boundary.

The weights involved in the shift are calculated by using a training set of data. This involves running sets of known-origin pairs through the system to obtain scores, resulting in a training model. In an ideal situation, one would have three databases upon which to build an FVC system; the background database (used to build a model of the distribution of the acoustic feature of interest), the development database (used to calculate the weights for logistic-regression calibration and for general optimisation), and the test database (previously unused recordings that can be used to test the system – often the offender and suspect recordings) (Morrison et al., 2012). In this study, due to the limitations in the amount of data, the calibration weights were obtained using a cross-validated procedure; each derived score was referenced against every other score in the database to produce the weights. This is quite a common technique, and it has been shown to work well with MVKD-based LR outputs (Morrison, 2009b; Morrison & Kinoshita, 2008; Morrison et al., 2011).

The FoCal toolkit[5] was used for logistic-regression calibration (Brümmer & du Preez, 2006).

3.3 Metrics of performance

Evidence must be reported alongside measures of *accuracy* (also *validity*) and *precision* (also *reliability*) in order to be admitted as scientific evidence in court (Morrison, 2009a, p. 299). Accuracy refers to the "closeness of agreement between a measured quantity value and a true quantity value of a measurand" (BIPM et al., 2008, p. 21), and precision refers to the "closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions" (BIPM et al., 2008, p. 22).

Two metrics that can be used to assess output performance under this requirement are the *log-likelihood-ratio cost* (the measure of validity) (Brümmer & du Preez, 2006), and *credible intervals* (the measure of reliability) (Morrison, 2011).

Log-likelihood-ratio cost

One way of assessing validity is to find the overall correct-classification rate of the output – the equal error rate (EER). However, EER is "based on a categorical thresholding, error versus not-error, rather than a gradient strength of evidence" (Morrison, 2011, p. 93). It is not an appropriate measure of system performance as it refers to posterior probability (a question of guilt or innocence). Furthermore, these "error versus not-error" decisions are binary, unlike LRs, which are continuous; "[t]he size of a likelihood ratio indicates the strength of its support for one hypothesis over the other" (Morrison, 2011, p. 93). EER does not provide any means of assessing the strength of the LRs of an output. So, while EER can be a useful metric for the overall discriminability of a system, it is not strictly appropriate for use in FVC.

It has been argued that a more appropriate metric for assessing the validity of an output is the log-likelihood-ratio cost (henceforth C_{llr}) (Brümmer & du Preez, 2006). C_{llr} can be calculated using 4).

$$C_{llr} = \frac{1}{2}\left(\frac{1}{\mathrm{N_{H_p}}}\sum_{\mathrm{i\ for\ H_p=true}}^{\mathrm{N_{H_p}}} \log_2\left(1+\frac{1}{\mathrm{LR_i}}\right) + \frac{1}{\mathrm{N_{H_d}}}\sum_{\mathrm{j\ for\ H_d=true}}^{\mathrm{N_{H_d}}} \log_2\left(1+\mathrm{LR_j}\right)\right) \qquad (4)$$

$\mathrm{N_{H_p}}$ and $\mathrm{N_{H_d}}$ refer to the numbers of SS and DS comparisons. $\mathrm{LR_i}$ and $\mathrm{LR_j}$ refer to the LRs derived from these SS and DS comparisons, respectively.

C_{llr} takes into account the magnitude of consistent-with-fact (and contrary-to-fact) LR values, and assigns them appropriate penalties. For example, $\log_{10}\mathrm{LR}= -5$ for an SS comparison would contribute a much heavier penalty to C_{llr} than $\log_{10}\mathrm{LR}= -0.5$ for an SS comparison. Similarly, a

[5] https://sites.google.com/site/nikobrummer/focal

correctly-classified SS comparison with $\log_{10}$LR=0.5 does not provide much support for the same-speaker hypothesis, and would therefore contribute a larger penalty than $\log_{10}$LR=4 for an SS comparison (Morrison, 2011, p. 94). For any output, an obtained C_{llr} value less than 1 implies that the output is providing a certain amount of information, and the validity gets better as C_{llr} approaches 0. The FoCal toolkit[6] was also used for calculating C_{llr} values in this study (Brümmer & du Preez, 2006).

Credible intervals

To assess reliability (precision), we used 95% credible intervals (95% *CI*). Credible intervals are "the Bayesian analogue of frequentist confidence intervals", and have the following interpretation: "we are 95% certain that the true value of the parameter we wish to estimate lies within the 95% credible interval" (Morrison, 2011, p. 95). In this study, uniform prior odds are assumed, so the actual calculations are identical to frequentist confidence intervals. It is also important to note that as there were only two recordings of each speaker, 95% *CI* values can only be estimated from the DS comparisons.

4 Results

Table 3 shows the best-performing outputs for each target word in terms of C_{llr}.

word	C_{llr}	**formant combination**	**DCT coefficients**	**95% CI**
Bye	0.158	23	5	9.996
Right	0.271	123	2	7.272
No	0.318	123	2	3.472
Left	0.342	123	2	4.249
Hello	0.392	123	2	3.518
Yes	0.527	23	5	4.232

Table 3: Best-performing outputs for each target word by C_{llr}.

Table 3 shows that "bye" performed best in terms of C_{llr}, and "yes" was the worst by the same measure. However, on closer inspection we see that the 95% *CI* for "bye" is poor in comparison to the other words. This is not a coincidence; "bye" consistently performed the best in terms of C_{llr} even with other combinations of the parameters, while performing the worst in terms of 95% *CI*.

A Pearson correlation test shows a negative correlation between the C_{llr} and 95% *CI* values (= -0.700; p < 0.0001) across all words. This is actually to be expected; Morrison (2011)) notes that one would ideally hope for low values for both metrics, but in practice, this is not often the case. It is clear that there is a trade-off when it comes to assessing the performance of the outputs.

When comparing the typical trajectories of the vowels in these words, it is noticeable that performance, in terms of C_{llr}, roughly corresponds to the level of dynamism of the trajectories. 2nd, 3rd, 4th, and 5th degree DCT-modelled curves tended to perform the best.

Presented in Figure 2 are the Tippett plots for the best-performing outputs of each word. Tippett plots show the cumulative distribution of $\log_{10}$LRs for SS and DS comparisons. As stated earlier, in a good output we expect most SS comparisons to produce $\log_{10}$LRs > 0, and most DS comparisons to produce $\log_{10}$LRs < 0. The counter-factual LRs (circled in Figure 2a as an example) that are "penalised" by C_{llr} (and their strength) become clear when inspecting a Tippett plot. The EER is also made clear in a Tippett plot; it is the crossing point of the SS and DS lines (indicated by the arrow in Figure 2e as an example). 95% *CI* bands (grey dotted curves) are also included in the Tippett plots given in Figure 2 for the DS comparison curves.

As can be seen in Figure 2, in all outputs, the DS LRs achieve greater values compared to the SS LRs; the DS curves are less steep than the SS curves. This is partly due to the number of DS comparisons (210) in each output outnumbering the number of SS comparisons (15). Also, when counter-factual, the DS comparisons tend to be more counter-factual than SS comparisons (except "yes" SS comparisons).

It is immediately obvious that "bye" is the highest performer; it achieves the greatest SS and DS values of all the outputs (values furthest away from $\log_{10}$LR $= 0$) and it has 100% correct discrimination for SS comparisons. It does produce misleading DS LRs, but the strength of these LRs is comparable with the other outputs. "No" also achieves 100% correct discrimination for SS comparisons, and "right" and "left" come very close to doing so.

[6] https://sites.google.com/site/nikobrummer/focal

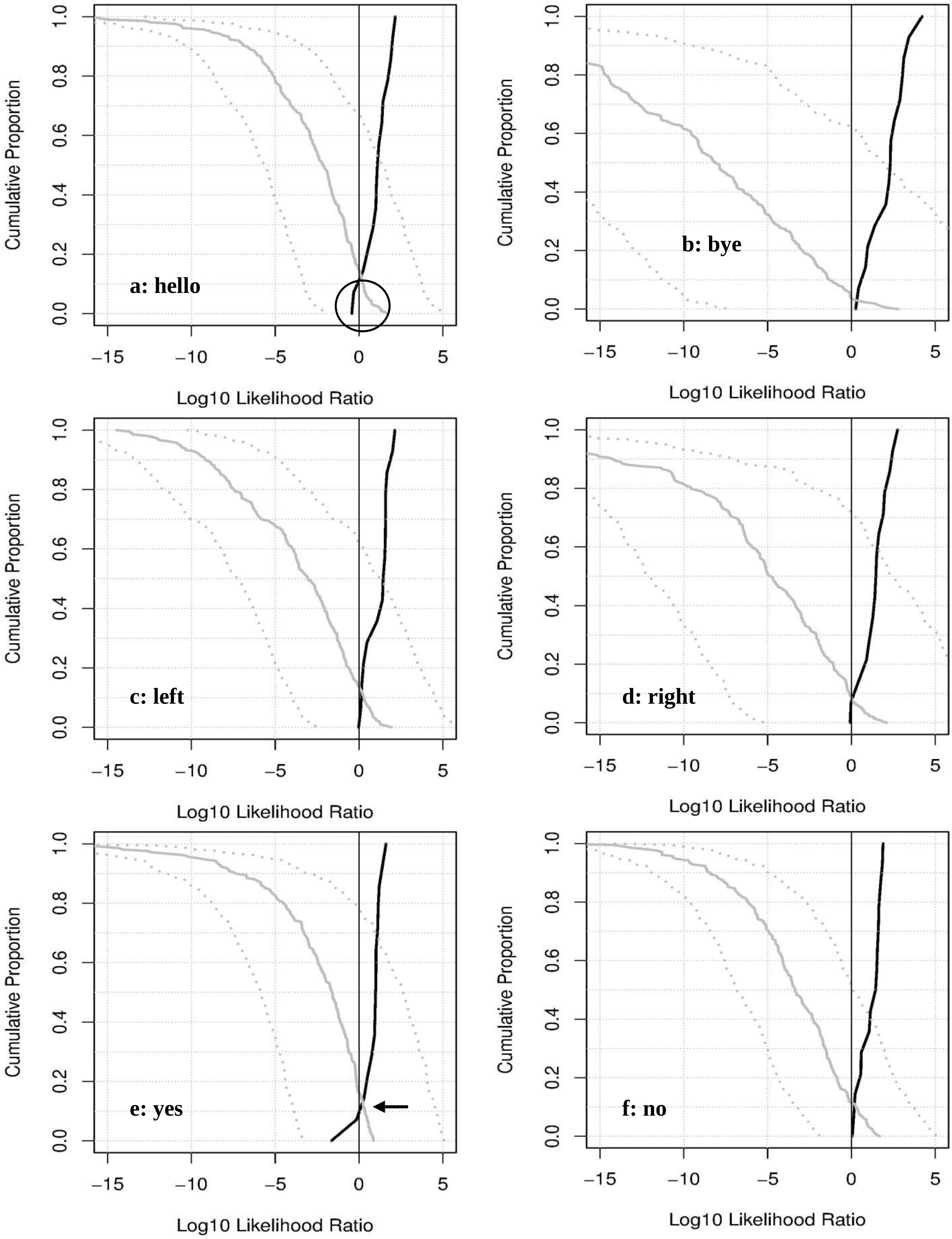


Figure 2: Tippett plots of the best-performing outputs for the target words. Black curve = calibrated SS LRs; Grey solid curve = calibrated DS LRs. Dotted grey curves = 95% *CI* band. The circle in Panel A indicates the counter-factual LRs and the arrow in Panel E indicates the EER.

5 Discussion

For the participants in this study, phonetic realisation varied greatly between speakers, and speakers were generally consistent internally. Table 4 and Table 5 show various phonetic realisations for “bye” and “hello” respectively. The effect of this variation is seen in the overall performance of our system; our DS comparisons tended to perform very well.

bye			
consonant	start target	length	end target
	a	unmarked	i
b	ɑ	˘	ɪ
p	æ	ˑ	e
	ɐ	ː	ə

Table 4: Various phonetic variations seen in the production of “bye”. (Not all combinations were realised– this is a list of articulations that appeared in the given positions.)

hello				
consonant	vowel	consonant	target 1	target 2
h	ɛ	l	ə	ʊ
	ə	ˡ	ɜ	u
	ɐ	ɾ	ɛ	
	a	Ø	o	

Table 5: Various phonetic variations seen in the production of “hello”. Another common final vowel was [oː].

While our research aim makes no mention of a comparison of our L2-based FVC system with similar traditional L1-based FVC systems, it is still an issue of particular interest. While it is not theoretically appropriate to directly compare C_{llr} values between systems unless the experimental settings are identical, doing so can provide a rough comparison of two systems. Morrison (2009b) looked at parametric representations (DCTs and polynomials) of the formant trajectories of five Australian English diphthongs, namely /aɪ/, /eɪ/, /oʊ/, /aʊ/, /ɔɪ/ (/aɪ/ corresponds to the /aɪ/ in this study, and /oʊ/ corresponds to the /əʊ/ in this study) from 27 Australian males. The best /aɪ/ output achieved a C_{llr} of 0.156, compared to 0.158 (“bye”) and 0.271 (“right”) in this study, and the best /oʊ/ (/əʊ/) output achieved 0.129, compared to 0.318 (“no”) and 0.392 (“hello”) in this study. We can see that the performance of the diphthong-specific outputs is quite comparable to the equivalent outputs in this study. This implies that L2-based FVC systems have no major shortcomings.

6 Conclusion

This study was the first to build an LR-based FVC system on L2 speech production, motivated by the relative prevalence of crimes involving L2 speakers. 15 native HK Cantonese-speaking males participated in the research. Six common words were targeted, and DCT-modelled parametric curves were fitted to the formant trajectories of the six target words. The coefficient values of the DCT-modelled curves were used as feature vectors in the LR calculations. The MVKD procedure (Aitken & Lucy, 2004) was used to produce LRs for each word. We used logistic-regression calibration (Brümmer & du Preez, 2006) to calibrate the outputs of the MVKD procedure.

Each output was evaluated with two metrics; the log-likelihood-ratio cost (C_{llr}) measured validity, and credible intervals (95% *CI*) measured reliability. We found that the words with more dynamic formant trajectories tended to perform best, and outputs involving F1, F2 and F3 performed better than outputs involving just F1 and F2, or F2 and F3. 2nd, 3rd, 4th, and 5th degree DCT-modelled curves tended to produce the best outputs.

In terms of the research question – whether or not FVC can be performed on L2 speech – we have clearly demonstrated that FVC can, and does, work on L2 speech. Further, we achieved results comparable to traditional L1-based FVC systems, which is certainly promising for the prospects of the field.

Acknowledgments

The authors greatly appreciate the very detailed comments of the three anonymous reviewers. This study is part of the first author’s honours thesis at the Australian National University.

References

Aitken, C. G. (1995). *Statistics and the evaluation of evidence for forensic scientists*. Chichester: J. Wiley.

Aitken, C. G., & Lucy, D. (2004). Evaluation of trace evidence in the form of multivariate data. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 53*(1), 109-122.

BIPM, I., IFCC, I., IUPAC, I., & ISO, O. (2008). Evaluation of measurement data—guide for the expression of uncertainty in measurement. JCGM 100: 2008.

Brümmer, N., & du Preez, J. (2006). Application-independent evaluation of speaker detection. *Computer Speech & Language, 20*(2), 230-275.

Cox, F. (2012). *Australian English pronunciation and transcription*. Cambridge: Cambridge University Press.

Durou, G. (1999). Multilingual text-independent speaker identification. *Proceedings of the Multi-Lingual Interoperability in Speech Technology*, 115-118.

Evett, I. (1998). Towards a uniform framework for reporting opinions in forensic science casework. *Science & Justice, 38*(3), 198-202.

Harrington, J., Cox, F., & Evans, Z. (1997). An acoustic phonetic study of broad, general, and cultivated Australian English vowels. *Australian Journal of Linguistics, 17*(2), 155-184.

Ishihara, S., & Kinoshita, Y. (2008). How many do we need? Exploration of the population size effect on the performance of forensic speaker classification. *Proceedings of the Interspeech 2008*, 1941-1944.

Kajarekar, S. S., Scheffer, N., Graciarena, M., Shriberg, E., Stolcke, A., Ferrer, L., & Bocklet, T. (2009). The SRI NIST 2008 speaker recognition evaluation system. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, 4205–4208.

Ladefoged, P., & Disner, S. F. (2012). *Vowels and Consonants*. Chichester: Wiley.

Li, J., & Rose, P. (2012). Likelihood ratio-based forensic voice comparison with F-pattern and tonal F0 from the Cantonese /ɔy/ diphthong. *Proceedings of the 14th Australasian International Conference on Speech Science and Technology*, 201-204.

Morrison, G. S. (2008). Forensic voice comparison using likelihood ratios based on polynomial curves fitted to the formant trajectories of Australian English /aɪ/. *International Journal of Speech Language and the Law, 15*, 247-264.

Morrison, G. S. (2009a). Forensic voice comparison and the paradigm shift. *Science & Justice, 49*(4), 298-308.

Morrison, G. S. (2009b). Likelihood-ratio forensic voice comparison using parametric representations of the formant trajectories of diphthongs. *The Journal of the Acoustical Society of America, 125*, 2387-2397.

Morrison, G. S. (2011). Measuring the validity and reliability of forensic likelihood-ratio systems. *Science & Justice, 51*(3), 91-98.

Morrison, G. S. (2013). Tutorial on logistic-regression calibration and fusion: Converting a score to a likelihood ratio. *Australian Journal of Forensic Sciences, 45*(2), 173-197.

Morrison, G. S., & Kinoshita, Y. (2008). Automatic-Type Calibration of Traditionally Derived Likelihood Ratios: Forensic Analysis of Australian English /o/ Formant Trajectories. *Proceedings of the Interspeech 2008*, 1501-1504.

Morrison, G. S., Ochoa, F., & Thiruvaran, T. (2012). Database selection for forensic voice comparison. *Proceedings of the Odyssey 2012*, 62-77.

Morrison, G. S., Zhang, C., & Rose, P. (2011). An empirical estimate of the precision of likelihood ratios from a forensic-voice-comparison system. *Forensic science international, 208*(1), 59-65.

Robertson, B., & Vignaux, G. A. (1995). *Interpreting evidence*. Chichester: Wiley.

Rose, P. (2002). *Forensic speaker identification*. London & New York: Taylor & Francis Forensic Science Series.

Rose, P. (2004). Technical forensic speaker identification from a Bayesian linguist's perspective. *Proceedings of the Odyssey 2004*, 3-10.

Rose, P. (2013). More is better: likelihood ratio-based forensic voice comparison with vocalic segmental cepstra frontends. *International Journal of Speech Language and the Law, 20*(1), 77-116.

Scheffer, N., Ferrer, L., Graciarena, M., Kajarekar, S., Shriberg, E., & Stolcke, A. (2011). The SRI NIST 2010 speaker recognition evaluation system. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)*, 5292-5295.

Similarity Metrics for Clustering PubMed Abstracts for Evidence Based Medicine

Hamed Hassanzadeh[†] **Diego Mollá**[◇] **Tudor Groza**[‡] **Anthony Nguyen**[♮] **Jane Hunter**[†]

[†]School of ITEE, The University of Queensland, Brisbane, QLD, Australia
[◇]Department of Computing, Macquarie University, Sydney, NSW, Australia
[‡]Garvan Institute of Medical Research, Darlinghurst, NSW, Australia
[♮]The Australian e-Health Research Centre, CSIRO, Brisbane, QLD, Australia

h.hassanzadeh@uq.edu.au, diego.molla-aliod@mq.edu.au
t.groza@garvan.org.au, anthony.nguyen@csiro.au, jane@itee.uq.edu.au

Abstract

We present a clustering approach for documents returned by a PubMed search, which enable the organisation of evidence underpinning clinical recommendations for Evidence Based Medicine. Our approach uses a combination of document similarity metrics, which are fed to an agglomerative hierarchical clusterer. These metrics quantify the similarity of published abstracts from syntactic, semantic, and statistical perspectives. Several evaluations have been performed, including: an evaluation that uses ideal documents as selected and clustered by clinical experts; a method that maps the output of PubMed to the ideal clusters annotated by the experts; and an alternative evaluation that uses the manual clustering of abstracts. The results of using our similarity metrics approach shows an improvement over K-means and hierarchical clustering methods using TF-IDF.

1 Introduction

Evidence Based Medicine (EBM) is about individual patients care and providing the best treatments using the best available evidence. The motivation of EBM is that clinicians would be able to make more judicious decisions if they had access to up-to-date clinical evidence relevant to the case at hand. This evidence can be found in scholarly publications available in repositories such as PubMed[1]. The volume of available publications is enormous and expanding. PubMed repository, for example, indexes over 24 million abstracts. As a result, methods are required to present relevant recommendations to the clinician in a manner that highlights the clinical evidence and its quality.

The EBMSummariser corpus (Mollá and Santiago-martinez, 2011) is a collection of evidence-based recommendations published in the *Clinical Inquiries* column of the *Journal of Family Practice*[2], together with the abstracts of publications that provide evidence for the recommendations. Visual inspection of the EBMSummariser corpus suggests that a combination of information retrieval, clustering and multi-document summarisation would be useful to present the clinical recommendations and the supporting evidence to the clinician.

Figure 1 shows the title (question) and abstract (answer) associated with one recommendation (Mounsey and Henry, 2009) of the EBM-Summariser corpus. The figure shows three main recommendations for treatments to hemorrhoids. Each treatment is briefly presented, and the quality of each recommendation is graded (A, B, C) according to the Strength of Recommendation Taxonomy (SORT) (Ebell et al., 2004). Following the abstract of the three recommendations (not shown in Figure 1), the main text provides the details of the main evidence supporting each treatment, together with the references of relevant publications. A reference may be used for recommending several of the treatments listed in the recommendations. Each recommendation is treated in this study as a cluster of references for evaluation purposes, and the corpus therefore contains overlapping clusters.

It has been observed that a simple K-means clustering approach provides a very strong base-

[1] www.ncbi.nlm.nih.gov/pubmed

[2] www.jfponline.com/articles/clinical-inquiries.html

Hamed Hassanzadeh, Diego Mollá, Tudor Groza, Anthony Nguyen and Jane Hunter. 2015. Similarity Metrics for Clustering PubMed Abstracts for Evidence Based Medicine . In *Proceedings of Australasian Language Technology Association Workshop*, pages 48−56.

Which treatments work best for Hemorrhoids?
Excision is the most effective treatment for thrombosed external hemorrhoids (strength of recommendation [SOR]: B, retrospective studies). For prolapsed internal hemorrhoids, the best definitive treatment is traditional hemorrhoidectomy (SOR: A, systematic reviews). Of nonoperative techniques, rubber band ligation produces the lowest rate of recurrence (SOR: A, systematic reviews).

Figure 1: Title and abstract of one sample (Mounsey and Henry, 2009) of the *Clinical Inquiry* section of *Journal of Family Practice*.

line for non-overlapping clustering of the EBM-Summariser corpus (Shash and Mollá, 2013; Ekbal et al., 2013). Past work was based on the clustering of the documents included in the EBMSummariser corpus. But in a more realistic scenario one would need to cluster the output from a search engine. Such output would be expected to produce much noisier data that might not be easy to cluster.

In this paper, we cluster documents retrieved from PubMed searches. We propose a hierarchical clustering method that uses custom-defined similarity metrics. We perform a couple of evaluations using the output of PubMed searches and the EBMSummariser corpus. Our results indicate that this method outperforms a K-means baseline for both the EBMSummariser corpus and PubMed's retrieved documents.

The remainder of the paper is structured as follows. Section 2 describes related work. Section 3 provides details of the clustering approach and the evaluation approaches. Section 4 presents the results, and Section 5 concludes this paper.

2 Related Work

Document clustering is an unsupervised machine learning task that aims to discover natural groupings of data and has been used for EBM in several studies. Lin and Demner-Fushman (2007) clustered MEDLINE citations based on the occurrence of specific mentions of interventions in the document abstracts. Lin et al. (2007) used K-means clustering to group PubMed query search results based on TF-IDF. Ekbal et al. (2013) used genetic algorithms and multi-objective optimisation to cluster the abstracts referred in the EBMSummariser corpus, and in general observed that it was difficult to improve on Shash and Mollá (2013)'s K-means baseline, which uses TF-IDF similar to Lin and Demner-Fushman (2007).

It can be argued that clustering the abstracts that are cited in the EBMSummariser corpus is easier than clustering those from Pubmed search results, since the documents in the corpus have been curated by experts. As a result, all documents are relevant to the query, and they would probably cluster according to the criteria determined by the expert. However, in a more realistic scenario the documents that need to be clustered are frequently the output of a search engine. Therefore, there might be documents that are not relevant, as well as duplicates and redundant information. An uneven distribution of documents among the clusters may also result.

There are several approaches to cluster search engine results (Carpineto et al., 2009). A common approach is to cluster the documents snippets (*i.e.,* the brief summaries appearing in the search results page) instead of the entire documents (Ferragina and Gulli, 2008). Our approach for clustering search engine results is similar to this group of approaches, since we only use the abstract of publications instead of the whole articles. The abstracts of scholarly publications usually contain the key information that is reported in the document. Hence, it can be considered that there is less noise in abstracts compared to the entire document (from a document clustering perspective). A number of clustering approaches can then be employed to generate meaningful clusters of documents from search results (Zamir and Etzioni, 1998; Carpineto et al., 2009).

3 Materials and Method

In this section we describe an alternative to K-means clustering over TF-IDF data. In particular, we devise separate measures of document similarity and apply hierarchical clustering using our custom matrix of similarities.

We first introduce the proposed semantic similarity measures for quantifying the similarity of abstracts. We then describe the process of preparing and annotating appropriate data for clustering semantically similar abstracts. Finally, the experimental set up will be explained.

Prior to describing the similarity measures, a glossary of the keywords that are used in this section is introduced:

Effective words: The words that have noun, verb, and adjective Part of Speech (POS) roles.

Effective lemmas: Lemma (canonical form) of effective words of an abstract.

Skipped bigrams: The pairs of words which are created by combining two words in an abstract that are located in arbitrary positions.

3.1 Quantifying similarity of PubMed abstracts

In order to be able to group the abstracts which are related to the same answer (recommendation) for a particular question, the semantic similarity of the abstracts was examined. A number of abstract-level similarity measures were devised to quantify the semantic similarity of a pair of abstracts. Since formulating the similarity of two natural language pieces of text is a complex task, we performed a comprehensive quantification of textual semantic similarity by comparing two abstracts from different perspectives. Each of the proposed similarity measures represents a different view of the similarity of two abstracts, and therefore the sum of all of them represents a combined view of each of these perspectives. The details of these measures can be found below. Note that all the similarity measures have a normalised value between zero (lowest similarity) and one (highest similarity).

Word-level similarity: This measure calculates the number of overlapping words in two abstracts which is then normalised by the size of the longer abstract (in terms of the number of all words). The words are compared in their original forms in the abstracts (even if there were multiple occurrences). Equation (1) depicts the calculation of Word-level Similarity (WS).

$$WS(A_1, A_2) = \frac{\sum_{w_i \in A_1} \begin{cases} 1 & \text{if } w_i \text{ is in } A_2 \\ 0 & \text{Otherwise} \end{cases}}{L} \tag{1}$$

where A_1 and A_2 refer to the bags of all words in two given abstracts (including multiple occurrences of words), and L is the size of the longest abstract in the pair.

Word's lemma similarity: This measure is calculated similarly to the previous measure, but the lemma of words from a pair of abstracts are compared to each other, instead of their original display forms in the text, using WordNet (Miller, 1995). For example, for a given pair of words, such as *criteria* and *corpora*, their canonical forms (*i.e.*, *criterion* and *corpus*, respectively) are looked up in WordNet prior to performing the comparison.

Set intersection of effective lemmas: The sets of lemmas of effective words of abstract pairs are compared. The number of overlapping words (or the intersection of two sets) is normalised by the size of the smaller abstract. In contrast to the previous measure, only unique effective lemmas participate in the calculation of this measure. This measure is calculated as follows:

$$SEL(A_1, A_2) = \frac{|A_1^{set} \cap A_2^{set}|}{S} \tag{2}$$

In Equation (2), A_1^{set} and A_2^{set} are the sets of effective lemmas of two abstracts, and S is the size of the smallest abstract in a pair.

Sequence of words overlap: We generate sliding windows of different sizes of words, from a window of two words up to the size of the longest sentence in a pair of abstracts. We compute the number of equal sequences of words of two abstracts (irrespective of length). Also, we keep the size of the longest equal sequence of words that the two abstracts share together. Hence, this results in two similarity measures; (*i*) the number of shared sequences of different sizes, and (*ii*) the size of the longest shared sequence. Due to the variety of sizes of sentences / abstracts and therefore varying sizes and number of sequences, we normalise each of these measures to reach a value between zero and one. In addition, following the same rationale, sequence-based measures are calculated by only considering effective words in abstracts, and alternatively, from a grammatical perspective, by only considering POS tags of the constituent words of abstracts. The number of shared sequences (or Shared Sequence Frequency — *SSF*) for two given abstracts (*i.e.*, A_1 and A_2) is calculated as follows:

$$SSF(A_1, A_2) = \frac{\sum_{l=2}^{M} \frac{\sum_{S_l \in A_1} \begin{cases} 1 & \text{if } S_l \in A_2 \\ 0 & \text{Otherwise} \end{cases}}{N}}{M} \tag{3}$$

In Equation (3), M is the size of the longest sentence in both abstracts and N is the number of available sequences (*i.e.*, S in formula) with size l.

POS tags sequence alignment: For this similarity measure, a sequence of the POS tags of words in an abstract is generated. The Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) was employed for aligning two sequences of POS tags from a pair of abstracts to find their similarity ratio. The Needleman-Wunsch algorithm is an efficient approach for finding the best alignment between two sequences, and has been successfully applied, in particular in bioinformatics, to measure regions of similarity in DNA, RNA or protein sequences.

Jaccard Similarity: An abstract can be considered as a bag of words. To incorporate this perspective, we calculate the Jaccard similarity coefficient of a pair of abstracts. We also calculate the Jaccard similarity of sets of effective lemmas of abstract pairs. The former similarity measure shows a very precise matching of the occurrences of words in exactly the same form (singular / plural, noun / adjective / adverb, and so on), while the latter measure considers the existence of words in their canonical forms.

Abstract lengths: Comparing two abstracts from a word-level perspective, the relative length of two abstracts in terms of their words (length of smaller abstracts over the longer one) provides a simple measure of similarity. Although this can be considered as a naive attribute of a pair of abstracts, it has been observed that this measure can be useful when combined with other more powerful measures (Hassanzadeh et al., 2015).

Cosine similarity of effective lemmas: In order to calculate the cosine similarity of the effective lemmas of a pair of abstracts, we map the string vector of the sequence of effective lemmas to its corresponding numerical vector. The numerical vector, with the dimension equal to the number of all unique effective lemmas of both abstracts, contains the frequency of occurrences of each lemma in the pair. For example, for the two sequences $[A, B, A, C, B]$ and $[C, A, D, B, A]$ the numerical vectors of the frequencies of the terms A, B, C and D for the sequences are $[2, 2, 1, 0]$ and $[2, 1, 1, 1]$, respectively. Equation (4) depicts the way the cosine similarity is calculated for two given abstracts A_1 and A_2.

$$Cosine(A_1, A_2) = \frac{V_1.V_2}{||V_1||||V_2||} \tag{4}$$

where V_1 and V_2 are the vector of lemmas of the effective words of two abstracts in a pair, and $V_1.V_2$ denotes the dot product of two vectors which is then divided by the product of their norms (*i.e.* $||V_1||||V_2||$).

Skipped bigram similarities: The set of the skipped bigrams of two abstracts can be used as a basis for similarity computation. We create the skipped bigrams of the effective words and then calculate the intersection of each set of these bigrams with the corresponding set from the other abstract in a pair.

3.2 Combining similarities

In order to assign an overall similarity score to any two given abstracts, the (non-weighted) average of all of the metrics listed above is calculated and is considered as the final similarity score. These metrics compare the abstracts from different perspectives, and hence, the combination of all of them results in a comprehensive quantification of the similarity of abstracts. This averaging technique has been shown to provide good estimation of the similarity of sentences when compared to human assessments both in general English and Biomedical domain corpora (Hassanzadeh et al., 2015).

3.3 Data set preparation and evaluation methods

In order to prepare a realistic testbed, we generated a corpus of PubMed abstracts. The abstracts are retrieved and serialised from the PubMed repository using E-utilities URLs[3]. PubMed is queried by using the 465 medical questions, unmodified, from the EBMSummariser corpus (Mollá and Santiago-martinez, 2011). The maximum number of search results is set to 20,000 (if any) and the results are sorted based on relevance using

[3] `www.ncbi.nlm.nih.gov/books/NBK25497/`

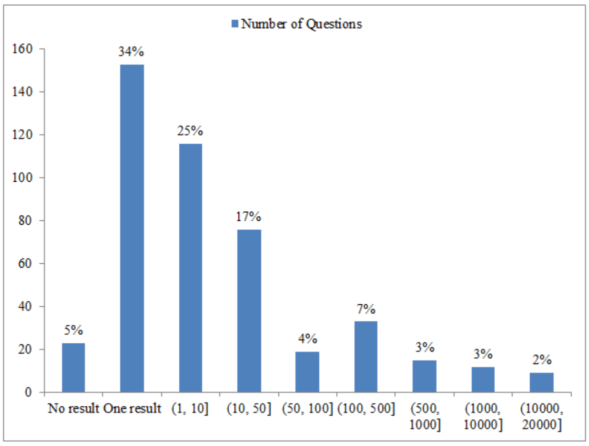


Figure 2: Statistics on the queried questions and their retrieved documents.

PubMed's internal relevance criteria.[4] In total, 212,393 abstracts were retrieved and serialised. The distributions of the retrieved abstracts per question were very imbalanced. There are a considerable number of questions with only one or no results from the PubMed search engine (39% of the questions). Figure 2 shows the frequency of the retrieved results and the number of questions with a given number and/or range of search results.

Some types of published studies may contain better quality of evidence than others, and some, such as opinion studies, provide very little evidence, if any at all. In addition, it is common to have a large number of search results for a given query. Hence, in order to find EBM-related publications as well as to ensure the quality and higher relevance of the abstracts, the retrieved abstracts were filtered based on their publication types. The types of publications are provided in the metadata returned by the PubMed abstracts. To determine the filters, we performed statistical analysis over available corpora in the EBM domain, in particular, EBMsummariser corpus (includes 2,658 abstracts), NICTA-PIBOSO corpus (includes 1,000 abstracts) (Kim et al., 2011), and our retrieved PubMed documents (includes 212,393 abstracts) — more details about the corpora can be found in Malmasi et al. (2015). Table 1 shows the frequency of the most frequent publication types in these EBM corpora. There are 72 different types of publications in PubMed[5], but we limited the retrieved abstracts to the seven more frequently occurring publication types in the EBM domain. Whenever we needed to reduce the number of retrieved abstracts from PubMed search results, we filter the results and only keep the abstracts with the mentioned publication types in Table 1. Note that each PubMed abstract can have more than one publication type. For example, a "Clinical Trial" abstract can also be a "Case Report" and so on. Hence, the sum of the percentages in Table 1 may exceed 100%. We assume that all the documents are informative when the number of returned search results is less than 50, and hence, no filtering was applied in these cases.

After retrieving the documents, in order to be able to evaluate the automatically-generated clusters of retrieved abstracts we devised two scenarios for generating gold standard clusters: Semantic Similarity Mapping and Manual Clustering.

Semantic Similarity Mapping scenario: We generated the gold standard clusters automatically using the cluster information from the EBMSummariser corpus. The answers for each question is known according to this corpus; each answer forms a cluster and citations associated with that answer are assigned to the respective cluster. In order to extend the gold standard to include all the retrieved PubMed abstracts, each abstract was assigned to one of these clusters. To assign an abstract to a cluster, we compute the similarity between the abstract and each of the cited abstracts for the question. To achieve this, we used our proposed combination of similarity measures. The abstract is assigned to the cluster with the highest average similarity. For example, suppose that for a given question there are three clusters of abstracts from the EBMSummariser corpus. By following this scenario, we assign each of the retrieved documents to one of these three clusters. We first calculate the average similarity of a given retrieved document to the documents in the three clusters. The cluster label (*i.e.,* 1, 2, or 3 in our example) for this given retrieved abstract is then adopted from the cluster with which it has the highest average similarity. This process is iterated to assign cluster labels to all the retrieved abstracts. However, it could occur that some clusters may not have any abstracts assigned to them. For the mentioned example, this will result when the retrieved documents would be assigned only to two of the three clusters. When that happens, the question is ignored to avoid a possible bias due to cluster

[4] www.nlm.nih.gov/pubs/techbull/so13/so13_pm_relevance.html

[5] www.ncbi.nlm.nih.gov/books/NBK3827/

Table 1: Statistics over the more common publication types in EBM domain corpora.

Publication Type	EBMSummariser	NICTA-PIBOSO	Retrieved
Clinical Trial	834 (31%)	115 (12%)	12,437 (6%)
Randomized Controlled Trial	763 (29%)	79 (8%)	13,849 (7%)
Review	620 (23%)	220 (22%)	26,162 (12%)
Comparative Study	523 (20%)	159 (16%)	19,521 (9%)
Meta-Analysis	251 (9%)	22 (2%)	2,067 (1%)
Controlled Clinical Trial	61 (2%)	9 (1%)	1,753 (1%)
Case Reports	37 (1%)	82 (8%)	8,599 (4%)

incompleteness. Following this scenario, we were able to create proper clusters for retrieved abstracts of 129 questions out of the initial 465.

Manual Clustering scenario: This scenario is based on the Pooling approach used in the evaluation of Information Retrieval systems (Manning et al., 2008). In this scenario, a subset of the top k retrieved documents is selected for annotation. To select the top k documents we use the above clusters automatically generated by our system. In order to be able to evaluate these automatically generated clusters, for each of them we determine its central document. A document is considered the central document of a cluster if it has the highest average similarity to all other documents in the same cluster. We then select the k documents that are most similar to the central document. The intuition is that if a document is close to the centre of a cluster, it should be a good representation of the cluster and it would less likely be noise. Two annotators (authors of this paper) manually re-clustered the selected top k documents following an annotation guideline. The annotators are not restricted to group the documents to a specific number of clusters (*e.g.*, to the same number of clusters as the EBMSummariser corpus). These manually generated clusters are then used as the gold standard clusters for the Manual Clustering evaluation scenario. The system is then asked to cluster the output of the search engine. Then, the documents from the subset that represents the pool of documents are evaluated against the manually curated clusters. The value of k in our experiment was set to two per cluster. In total, 10 queries (with different numbers of original clusters, from 2 to 5 clusters) were assessed for a total of 62 PubMed abstracts.

3.4 Experimental setup

We employed a Hierarchical Clustering (HC) algorithm in order to cluster the retrieved abstracts (Manning et al., 2008). HC methods construct clusters by recursively partitioning the instances in either a top-down or a bottom-up fashion (Maimon and Rokach, 2005). A hierarchical algorithm, such as Hierarchical Agglomerative Clustering (HAC), can use as input any similarity matrix, and is therefore suitable for our approach in which we calculate the similarity of documents from different perspectives.

As a baseline approach, we use K-means clustering (KM) with the same pre-processing as reported by Shash and Mollá (2013), namely we used the whole XML files output by PubMed and removed punctuation and numerical characters. We then calculated the TF-IDF of the abstracts, normalised each TF-IDF vector by dividing it by its Euclidean norm, and applied K-means clustering over this information. We employed the HC and KM implementations available in the *R* package (R Core Team, 2015).

We use the Rand Index metric to report the performance of the clustering approaches. Rand Index (RI) is a standard measure for comparing clusterings. It measures the percentage of clustering decisions on pairs of documents that are correct (Manning et al., 2008). Eq. 5 depicts the calculation of RI.

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

A true positive (*TP*) refers to assigning two similar documents to the same cluster, while a true negative (*TN*) is a decision of assigning two dissimilar documents to different clusters. A false positive (*FP*) occurs when two dissimilar docu-

Table 2: Clustering results over 129 questions of the EBMSummariser corpus.

Method	Rand Index
KM + TF-IDF	0.5261
HC + TF-IDF	0.5242
HC + Similarity Metrics	**0.6036***

* Statistically significant (p-value< 0.05) when compared with second best method.

ments are grouped into the same cluster. A false negative (*FN*) decision assigns two similar documents to different clusters.

4 Experimental Results

In this section, the results from applying our similarity metrics in order to cluster abstracts in the EBM domain are presented. We first introduce our experiments on clustering the abstracts from the EBMSummariser corpus and then we report the results over the retrieved abstracts from PubMed.

4.1 Results on EBMSummariser corpus

In order to evaluate our clustering approach using our similarity metrics, we first employ the EBMSummariser corpus. As previously mentioned, this corpus contains a number of clinical inquiries and their answers. In each of these answers, which are provided by medical experts, one or more citations to published works are provided with their PubMed IDs. We apply our clustering approach to group all the citations mentioned for a question and then compare the system generated clusters with those of the human experts. Table 2 shows the results of using Hierarchical Clustering (HC) and K-means clustering (KM) using the proposed similarity measures and TF-IDF information. In order to have a consistent testbed with our experiments over retrieved documents, the reported results of the corpus are over a subset of the available questions of the EBMSummariser corpus, that is, those 129 questions which were found valid for evaluation in the Semantic similarity mapping scenario in Section 3.3.

Note the improvement of the Rand Index against the TF-IDF methods, *i.e.*, 0.0775. This difference between HC using our similarity metrics and the next best approach, namely KM clustering using TF-IDF, is statistically significant (Wilcoxon signed rank test with continuity correction; p-value = 0.01092).

Our implementation of KM used 100 random starts. It should also be noted that KM can not be used over our similarity metrics, because the final representation of these metrics are the quantification of the similarity of a *pair* of documents and not a representation of a single document (*i.e.*, the appropriate input for KM clustering).

4.2 Results on PubMed documents

As mentioned in Section 3.3, we devised two methods for evaluating the system's generated clusters: the manual scenario, and the semantic similarity mapping scenario. The results of the clustering approach are reported for these two scenarios in Table 3 and Table 4, respectively.

Table 3 shows the results for the manual evaluation. It reports the comparison of the system's results against the manually clustered abstracts from the two annotators. This evaluation scenario shows that, in most cases, the HC approach that employs our similarity metrics produced the best Rand Index. The only exception occurs over the Annotator 1 clusters, where KM using TF-IDF gained better results (*i.e.*, 0.4038 RI). However, for this exception, it is noticed that this difference between the HC approach that uses our similarity metrics and KM using TF-IDF is not statistically significant (p-value=0.5).

Table 3 also shows that the results are similar for two of the three approaches on each annotator, which suggests close agreement among annotators. Note, incidentally, that the annotations were of *clusters*, and not of *labels*, and therefore standard inter-annotator agreements like Cohen's Kappa cannot be computed.

Table 4 shows the results of the methods by using the semantic similarity mapping evaluation approach. It can be observed that, similar to the manual evaluation scenario, HC clustering with the similarity metrics gained the best Rand Index. Finally, although the absolute values of Rand Index are much higher than that from the manual clustering evaluations, the difference between HC on our similarity metrics and the HC and KM methods on TF-IDF information is not statistically significant (p-value=0.1873).

To compare with the results reported in the literature, we computed the weighted mean cluster Entropy for the entire set of 456 questions. Ta-

Table 3: Clustering results over retrieved PubMed documents with Manual Clustering evaluation scenario (Rand Index) for 129 questions from the EBMSummariser corpus.

Methods	Annotator 1 clusters	Annotator 2 clusters	Average
KM + TF-IDF	0.4038	0.3095	0.3566
HC + TF-IDF	0.2877	0.2898	0.2887
HC + Similarity Metrics	0.3825	0.3926	**0.3875**

Table 4: Clustering results over retrieved PubMed documents with Semantic Similarity Mapping evaluation scenario for 129 questions from the EBMSummariser corpus.

Method	Rand Index
KM + TF-IDF	0.5481
HC + TF-IDF	0.5463
HC + Similarity Metrics	**0.5912**

Table 5: Clustering results over the entire EBM-Summariser corpus.

Method	Entropy
KM + TF-IDF (as in Shash and Mollá (2013))	0.260
KM + TF-IDF (our replication)	0.3959
HC + Similarity metrics	**0.3548***

* Statistically significant (p-value< 0.05) when compared with preceding method.

ble 5 shows our results and the results reported by Shash and Mollá (2013). The entropy generated by the HC system using our similarity metrics was a small improvement (lower entropy values are better) on the KM baseline (our replication of K-means using TF-IDF), which is statistically significant (p-value=0.00276). However, we observe that our KM baseline obtains a higher entropy than that reported in Shash and Mollá (2013), even though our replication would have the same settings as their system. Investigation into the reason for the difference is beyond the scope of this paper.

5 Conclusion

In this paper we have presented a clustering approach for documents retrieved via a set of PubMed searches. Our approach uses hierarchical clustering with a combination of similarity metrics and it reveals a significant improvement over a K-means baseline with TF-IDF reported in the literature (Shash and Mollá, 2013; Ekbal et al., 2013).

We have also proposed two possible ways to evaluate the clustering of documents retrieved by PubMed. In the semantic similarity mapping evaluation, we automatically mapped each retrieved document to a cluster provided by the corpus. In the manual clustering evaluation, we selected the top k documents and manually clustered them to form the annotated clusters.

Our experiments show that using semantic similarity of abstracts can help gain better clusters of related published studies, and hence, can provide an appropriate platform to summarise multiple similar documents. Further research will focus on employing domain-specific concepts in similarity metrics calculation as well as using tailored NLP tools in biomedical domain, such as BioLemmatizer (Liu et al., 2012). Further investigations can also be performed in order to track the effects and contribution of each of the proposed similarity measures on formulating the abstract similarities, and hence, on their clustering. In addition, in order to have more precise quantification of the similarity of abstracts, their sentences can be firstly classified using EBM related scientific artefact modeling approaches (Hassanzadeh et al., 2014). Knowing the types of sentences, the similarity measures can then be narrowed to sentence-level metrics by only comparing sentences of the same type. These investigations can be coupled with the exploration of overlapping clustering methods for allowing the inclusion of a document in several clusters.

Acknowledgments

This research is funded by the Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) – DE120100508. It is also partially supported by CSIRO Postgraduate Studentship and a visit of the first author to Macquarie University.

References

Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. 2009. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):17:1–17:38.

Mark H Ebell, Jay Siwek, Barry D Weiss, Steven H Woolf, Jeffrey Susman, Bernard Ewigman, and Marjorie Bowman. 2004. Strength of Recommendation Taxonomy (SORT): a Patient-Centered Approach to Grading Evidence in the Medical Literature. *The Journal of the American Board of Family Practice / American Board of Family Practice*, 17(1):59–67.

Asif Ekbal, Sriparna Saha, Diego Mollá, and K. Ravikumar. 2013. Multiobjective Optimization for Clustering of Medical Publications. In *Proceedings ALTA 2013*.

P. Ferragina and A. Gulli. 2008. A personalized search engine based on web-snippet hierarchical clustering. *Software: Practice and Experience*, 38(2):189–225.

Hamed Hassanzadeh, Tudor Groza, and Jane Hunter. 2014. Identifying scientific artefacts in biomedical literature: The evidence based medicine use case. *Journal of Biomedical Informatics*, 49:159–170.

Hamed Hassanzadeh, Tudor Groza, Anthony Nguyen, and Jane Hunter. 2015. A supervised approach to quantifying sentence similarity: With application to evidence based medicine. *PLoS ONE*, 10(6):e0129392, 06.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support Evidence Based Medicine. *BMC Bioinformatics*, 13(Suppl 2):S5.

Jimmy J. Lin and Dina Demner-Fushman. 2007. Semantic clustering of answers to clinical questions. In *AMIA Annual Symposium Proceedings*, volume 33, pages 63–103.

Yongjing Lin, Wenyuan Li, Keke Chen, and Ying Liu. 2007. A Document Clustering and Ranking System for Exploring {MEDLINE} Citations. *Journal of the American Medical Informatics Association*, 14(5):651–661.

Haibin Liu, Tom Christiansen, and William A. Baumgartner Karin Verspoor. 2012. Biolemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of Biomedical Semantics*, 3(3).

Oded Maimon and Lior Rokach. 2005. *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Shervin Malmasi, Hamed Hassanzadeh, and Mark Dras. 2015. Clinical Information Extraction Using Word Representations. In *Proceedings of the Australasian Language Technology Workshop (ALTA)*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

George A. Miller. 1995. Wordnet – a Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Diego Mollá and Maria Elena Santiago-martinez. 2011. Development of a corpus for evidence based medicine summarisation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*.

Anne L Mounsey and Susan L Henry. 2009. Clinical inquiries. Which treatments work best for hemorrhoids? *The Journal of family practice*, 58(9):492–3, September.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

R Core Team, 2015. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

SaraFaisal Shash and Diego Mollá. 2013. Clustering of medical publications for evidence based medicine summarisation. In Niels Peek, Roque Marn Morales, and Mor Peleg, editors, *Artificial Intelligence in Medicine*, volume 7885 of *Lecture Notes in Computer Science*, pages 305–309. Springer Berlin Heidelberg.

Oren Zamir and Oren Etzioni. 1998. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 46–54. ACM.

Finding Names in Trove: Named Entity Recognition for Australian Historical Newspapers

Sunghwan Mac Kim*
Data61, CSIRO,
Sydney, Australia
Mac.Kim@csiro.au

Steve Cassidy
Department of Computing
Macquarie University
Sydney, Australia
Steve.Cassidy@mq.edu.au

Abstract

Historical newspapers are an important resource in humanities research, providing the source materials about people and places in historical context. The Trove collection in the National Library of Australia holds a large collection of digitised newspapers dating back to 1803. This paper reports on some work to apply named-entity recognition (NER) to data from Trove with the aim of supplying useful data to Humanities researchers using the HuNI Virtual Laboratory. We present an evaluation of the Stanford NER system on this data and discuss the issues raised when applying NER to the 155 million articles in the Trove archive. We then present some analysis of the results including a version published as Linked Data and an exploration of clustering the mentions of certain names in the archive to try to identify individuals.

1 Introduction

In recent years, digitised newspaper archives have appeared on the web; they make fascinating reading but also provide important primary sources for historical research. The Trove (Holley, 2010)[1] Newspaper collection at the National Library of Australia (NLA) provides an interface for users to search and browse the collections of scanned pages using an optical character recognition (OCR) based transcript of each article. While the OCR results contain errors, they provide enough detail to enable a full-text index to return relevant results to search terms. The documents stored in the Trove archive are made freely available for any purpose by the National Library of Australia.

An abundance of natural language processing (NLP) tools have been developed for Digital Humanities (Brooke et al., 2015; Scrivner and Kübler, 2015) and such tools can greatly facilitate the work of Humanities scholars by automatically extracting information relevant to their particular needs from large volumes of historical texts. This project explores the use of Named Entity Recognition on the Trove Newspaper text to provide a resource for Humanities scholars.

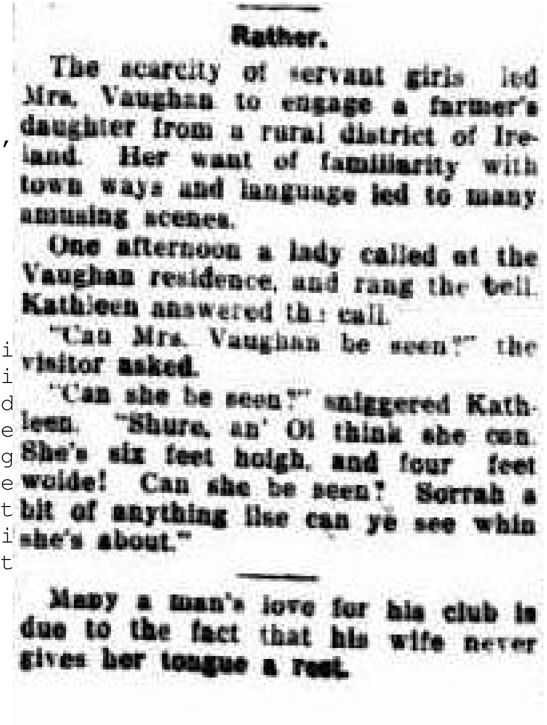
Newspapers are an important repository for historical research. Digitisation of newspaper text via Optical Character Recognition (OCR) enhances access and allows full text search in the archive. It also supports more sophisticated document processing using Natural Language Processing (NLP) techniques. Europe and the United States have actively participated in research on digitised historical newspapers and developed web-based applications using NLP to provide visualisation of useful information (Willems and Atanassova, 2015; Torget et al., 2011). The web-based applications have empowered digital humanities scholars to efficiently exploit historical newspaper content. The Europeana Newspapers project was performed to provide access to digitised historical newspapers from 23 European libraries (Willems and Atanassova, 2015). They used 10 million newspaper articles produced by OCR and a number of tools were developed for researchers. In particular, named entity recognition (NER) was applied to extract names of persons, places and organisations from the digitised newspapers. The University of North Texas and Stanford University used NER and topic modelling on 1 million digitised newspaper articles (Torget et al., 2011). They built interactive visualisation tools to provide researchers with the ability to find language patterns

*The work was done while the first author was a research associate at Macquarie University and was supported by a grant from NeCTAR.

[1] http://trove.nla.gov.au/

Sunghwan Mac Kim and Steve Cassidy. 2015. Finding Names in Trove: Named Entity Recognition for Australian Historical Newspapers . In *Proceedings of Australasian Language Technology Association Workshop*, pages 57−65.

```
{
 "id":"64154501",
 "titleId":"131",
 "titleName":"The Broadford Courier (Broadford,
 "date":"1917-02-02",
 "firstPageId":"6187953",
 "firstPageSeq":"4",
 "category":"Article",
 "state":["Victoria"],
 "has":[],
 "heading":"Rather.",
 "fulltext":"Rather. The scarcity of servant gi
 engage a farmer’s daughter from a rural distri
 of familiarity with town ways and language led
  One afternoon a lady called at the Vaughan re
   Kathleen answered the call.’ \"Can Mrs. Vaug
   asked. \"Can she be seen?\" sniggered Kathle
   she can. She’s six feet hoigh, and four feet
   Sorrah a bit of anything ilse can ye see whi
   man’s love for his club is due to the fact t
   gives her tongue a rest",
 "wordCount":118,
 "illustrated":false
 }
```



Rather.

The scarcity ol servant girls led Mrs. Vaughan to engage a farmer's daughter from a rural district of Ireland. Her want of familiarity with town ways and language led to many amusing scenes.

One afternoon a lady called at the Vaughan residence, and rang the bell. Kathleen answered th: call.

"Can Mrs. Vaughan be seen?" the visitor asked.

"Can she be seen?" sniggered Kathleen. "Shure, an' Oi think she can. She's six feet hoigh, and four feet woide! Can she be seen? Sorrah a bit of anything ilse can ye see whin she's about."

Many a man's love for his club is due to the fact that his wife never gives her tongue a rest.

Figure 1: An example Trove news article showing the JSON representation overlaid with an image of the original scanned document, taken from http://trove.nla.gov.au/ndp/del/article/64154501

embedded in the newspapers for any particular location or time period.

The HuNI-Alveo project is a collaborative research project among researchers at Deakin University, University of Western Australia and Macquarie University. The aim of the project is to ingest Trove digitised newspapers into Alveo[2] virtual lab and to build Alveo's NER functionality to provide Trove-derived person or location names for ingestion into HuNI[3] virtual lab. To reach the second goal, we use the Stanford NER system (Finkel et al., 2005). A significant challenge in this project is to process the large number of news articles (approximately 152 million). We are not aware of any other work that applies NER to a collection of this size (the Europeana project (Willems and Atanassova, 2015) is of a similar size but there are no published NER results on the whole collection).

The remainder of this paper is organised as follows. In Section 2 we discuss our dataset and lexical resources that are used in the NER task. Section 3 represents evaluation results of the Stanford NER systems and Section 4 describes the NER pipeline that we implemented. Then, a series of interesting results are presented and analysed in Section 5. Section 6 describes how the results of the NER process are published as linked data on the web. Finally, conclusions and directions for future work are given in Section 7.

2 Data

The central ideas in the HuNI-Alveo project are to apply an NER model to historical newspapers to allow humanities researchers to exploit automatically identified person or location names. We use the following resources in this work.

2.1 Trove

Trove[4] is the digital document archive of the National Library of Australia (Holley, 2010) and contains a variety of document types such as books, journals and newspapers. The newspaper archive in Trove consists of scanned versions of each page as PDF documents along with a transcription generated by ABBYY FineReader[5], which is is a state-of-the-art commercial optical character recognition (OCR) system. OCR is inherently

[2] http://alveo.edu.au/

[3] https://huni.net.au/

[4] http://trove.nla.gov.au/

[5] http://www.abbyy.com

error-prone and the quality of the transcriptions varies a lot across the archive; in particular, the older samples are of poorer quality due to the degraded nature of the original documents. Generally errors consist of poorly recognised characters leading to mis-spelling or just random text in some cases. Article boundary detection seems to be very good.

To help improve the quality of the OCR transcriptions, Trove provides a web based interface to allow members of the public to correct the transcriptions. This crowdsourcing approach produces a large number of corrections to newspaper texts and the quality of the collection is constantly improving. As of this writing, the Trove website reports a total of 170 million corrections to newspaper texts[6].

As part of this project, a snapshot sample of the Trove newspaper archive will be ingested into the Alveo Virtual Laboratory (Cassidy et al., 2014) for use in language research. One motivation for this is to provide a *snapshot* archive of Trove that can be used in academic research; this collection won't change and so can be used to reproduce published results. Alveo also aims to provide access to the data in a way that facilitates automatic processing of the text rather than the document-by-document interface provided by the Trove web API.

The snapshot we were given of the current state of the collection contains around 152 million articles from 836 different newspaper titles dating from between 1803 and 1954. The collection takes up 195G compressed and was supplied as a file containing the document metadata encoded as JSON, one document per line. A sample document from the collection is shown in Figure 1 along with an image of the original page.

2.2 HuNI

The HuNI Virtual Laboratory (Humanities Networked Infrastructure `http://huni.net.au`) supports researchers in the Humanities to discover, document and link records about people, places and events in Australia. HuNI harvests data from many Australian cultural websites into a single data store.

One of the goals of this project was to provide HuNI with a new dataset linking names to articles in Trove. To facilitate this, HuNI provided an export of their current list of person records, around 288,000 records. Our goal was to find mentions of these people in Trove, rather than finding all names which we thought would be too large a data set. While each person record contains a list of attributes such as occupation and biography along with first-name/last-name pair, only a small fraction of records have both first name and last name. We built a name dictionary by extracting the names of persons who have both first and last names, leaving a total of 41,497 names.

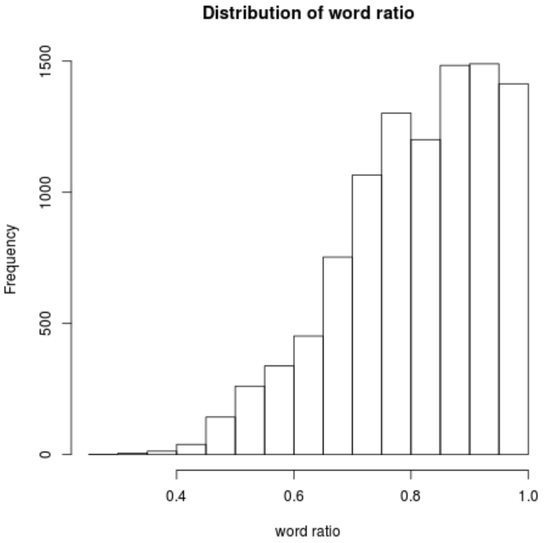


Figure 2: Histogram for the ratio of words to non-words over 10000 articles. The x-axis denotes the word frequency ratio and the y-axis denotes the number of articles.

2.3 Data Quality

As mentioned above, the quality of the OCR transcriptions in Trove is quite variable and we were concerned that the number of errors might be too high to allow useful results to be obtained from automatic processing. We thus investigate the quality of Trove in terms of word ratio with respect to a reference word list. The word list is derived from an Australian English dictionary [7] combined with the HuNI name list described above. Given an article, the word ratio is computed by dividing the number of words found in the dictionary by the total number of words in the article. This measures the relative frequency of words and non-words in

[6] `http://trove.nla.gov.au/system/stats?env=prod&redirectGroupingType=island#links`

[7] Derived from the Australian Learners Dictionary, available from `https://github.com/stevecassidy/ald`

the text . We assume that we can use word ratio as a proxy for OCR error rate and hence the quality of the text in each article (of course, many uncommon words will also be missing from the dictionary, making this measure an under-estimate of OCR error rate). Articles of poor quality would give a low word ratio, whereas articles of good quality would have high word ratio.

To evaluate the data, we randomly select 10000 articles[8] from the entire Trove dataset and estimated word frequency ratios over them. The histogram in Figure 2 shows the frequency ratio of words over 10000 articles. The x-axis denotes the frequency ratio of words and the y-axis denotes the number of new articles. We can observe the skew to the right in this small sample data which could indicate that the quality of the Trove data is not too bad. For instance, more than half the articles have a word frequency ratio greater than 0.8.

3 Evaluation

In this section we perform a comparative evaluation of two Stanford NER systems because we should make a decision about whether to train the NER system or not. To this end, we compare the performance of pre-trained Stanford NER with that of Stanford NER trained on our own training data. However, annotating data is a time-consuming and labour-intensive work and we thus use a semi-supervised learning approach. More specifically, training data is automatically generated using the pre-trained Stanford NER for randomly selected 600 articles and the produced silver standard data is used to train custom models for the Stanford NER system[9].

Some articles have a few sentences, even no names and they are not suitable for our evaluation. For this reason, we use the word frequency ratio described in Section 2.3 as a threshold to filter out inappropriate new articles. We randomly select 50 news articles from Trove that are not part of our training data using a word ratio threshold of 0.8. These articles were manually annotated using the MITRE annotation toolkit[10] to produce gold-standard test data for our evaluation.

On this test data, we evaluate the two Stanford NER systems and the comparison results are shown in Tables 1a and 1b. We can see that these two NER systems are on par with each other particularly in terms of F1 with respect to Person and Location, and our own trained Stanford NER does not provide any benefit. It would probably more desirable to use Stanford NER trained on more historical newspapers. However, this would be a labour-intensive and time-consuming task due to the huge amount of unannotated data. For these reasons, we use the pre-trained Stanford NER system, which gives us F1 scores of 0.76 for both person and location, in the rest of this paper.

As an aside, we also wondered if just using the HuNI supplied name list to look up names in the target articles would be a reasonable strategy. We ran an evaluation where words in target articles that were in the name list were tagged as PERSON instances. As might be expected with this approach, the recall is reasonable (0.75) since most of the target names will be found – errors are due to names not being present in the HuNI list. The precision though is very poor (0.07) since no cues are being used to differentiate ordinary words from names; hence, every occurrence of 'Carlton', 'rose' or 'brown' would count as a PERSON instance.

While we extracted and evaluated locations from the text, this paper concentrates on the use of person names. We hope to report on the application of location information in later work.

4 Extraction of Names

The goal of this work is to automatically extract person names and location names along with their relevant metadata from Trove. We use the pre-trained Stanford NER system that was evaluated in Section 3. The extraction of person names and their meta data is performed in four streaming steps as follows[11]:

1. Read news articles in Trove
2. Extract person entities from news context
3. Remove person names not found in the HuNI dictionary

[8] Actual number of articles is 9963 since 37 articles only have head information without article texts.

[9] We made a preliminary evaluation of Stanford NER given the increasing sizes of training data. We did not obtain any benefit from using more than 500 articles.

[10] http://mat-annotation.sourceforge.net/

[11] A noisy channel model was implemented to correct spelling errors in Trove but we did not obtain better quality texts using it. Furthermore, it seemed to be infeasible to apply it to the whole amount of Trove data due to extremely long processing time.

Entity	Precision	Recall	F1
Location	0.84	0.70	**0.76**
Organisation	0.56	0.47	0.51
Person	0.71	0.81	**0.76**
Totals	0.73	0.70	0.71

(a) Performance of pre-trained Stanford NER.

Entity	Precision	Recall	F1
Location	0.84	0.63	**0.72**
Organisation	0.54	0.28	0.37
Person	0.70	0.75	**0.73**
Totals	0.72	0.61	0.67

(b) Performance of Stanford NER trained on 600 articles.

Table 1: Performance comparison of Stanford NER systems in terms of precision, recall and f-score, figures quoted are micro-averaged.

4. Write tagged person named entities to a file

One of the most challenging issues in this work is to process large amounts of news articles, approximately 152 million articles as mentioned in Section 2.1. To tackle this issue, we implemented the extraction pipeline using a multiple threads to speed up processing. One extraction pipeline consists of several dedicated threads for reading, tagging and writing. In particular, multiple threads for tagging are used to communicate with multiple Stanford NER instances in a pipeline and this architecture leads to fast processing of large amounts of text. We utilised 15 virtual machines on the NeCTAR Research Cloud[12]; each machine was an m2.xlarge configuration with 48GB RAM and 12 virtual CPUs and the pipeline model ran on each virtual machine. The Trove data was divided into 30 chunks, each containing around 5 million news articles. Processing each chunk took 36 hours of processing time on average and the total processing time was about 72 hours.

The results contained 27 million person name mentions in 17 million articles; there were 731,673 different names - this includes some duplicates with different capitalisation.

Table 2 shows an example of the result for a name mention from Trove using the Stanford NER system; this includes some meta-data about the article containing the mention and a short text snippet showing the context of the first mention of the name in the article.

5 Results and Analysis

This section shows some fundamental and interesting results and analysis[13] obtained from our NER system. The main aim of our project is to foster research on digital humanities through the use of NER and to deliver all necessary results for digital humanities scholars. The following sections describe several results that could be interesting and important topics for digital humanists working with historical texts.

name: James Morgan,
article_id: 13977910
article_date: 1894-11-30,
article_source: The Sydney Morning Herald
(NSW : 1842 - 1954),
article_title: LEGISLATIVE ASSEMBLY.
THURSDAY, NOVEMBER 29.,
article_context: ...n standing in tho name of Mr.
James Morgan for tho appointment of a sole...,

Table 2: Extracted information for a person *James Morgan*.

5.1 Identifying Individuals

An important point to make here is that we are extracting *names* from the data, not *people*, however it is people that are of interest to Humanities researchers. Names are shared between many individuals over time as can be seen in Figure 3 which plots the occurence of the names of some Australian Prime Ministers for each year. Taking *Joseph Lyons* (red) as an example, there is a large peak in mentions around 1910 and a second peak in the 1930s. While these could refer to the same person, a little investigation shows that many of the 1910 mentions (eg. `http://trove.nla.gov.au/ndp/del/article/149796638`) refer to a Joseph Lyons arrested for destroying a railway line near Broken Hill (Figure 4). To make this data more useful to Humanities researchers it would be useful to be able to automatically cluster individuals within the data. This section describes one experiment in clustering names based on the in-

[12] `https://www.nectar.org.au/`

[13] Our results are publicly available via `http://trove.alveo.edu.au/` and we can perform a more detailed analysis using a query interface in SPARQL.

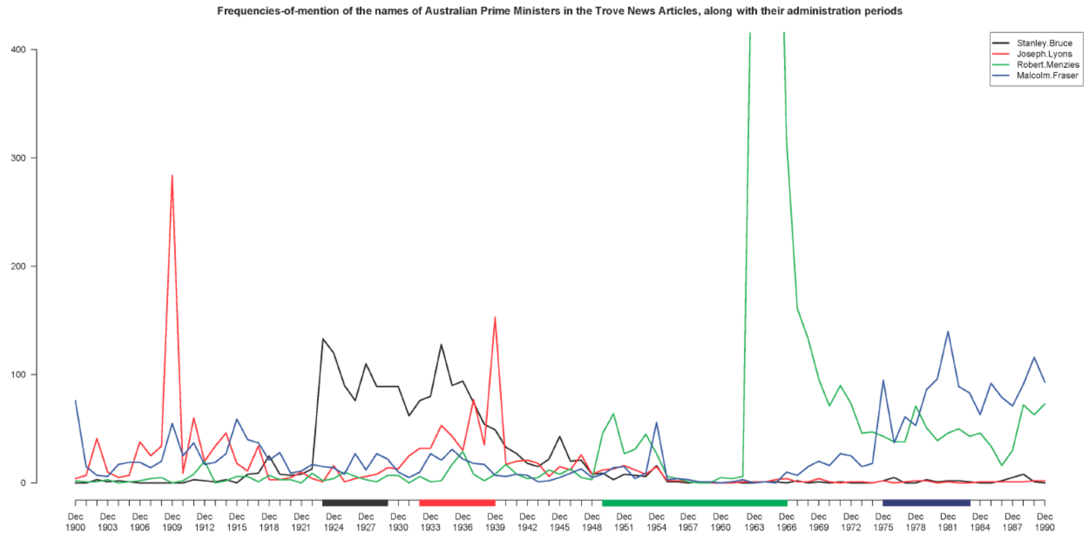
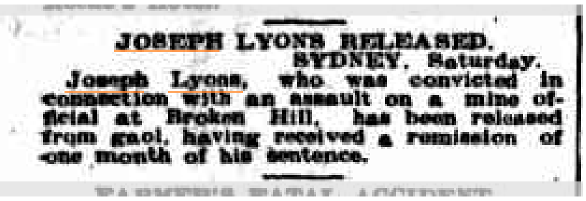


Figure 3: Frequencies-of-mention of the names of four Australian Prime Minsters along with their periods in office. Each colour zone on x-axis indicates the administration period of each Prime Minster.



JOSEPH LYONS RELEASED.

SYDNEY, Saturday.

Joseph Lyons, who was convicted in connection with an assault on a mine official at Broken Hill, has been released from gaol, having received a remission of one month of his sentence.

Figure 4: A mention of *Joseph Lyons* in 1909 which does not refer to the future Prime Minister (who was elected to the Tasmanian parliament in that year).

formation in the documents that mention them.

In this work we use a clustering approach on continuous vector space simply to distinguish whether the name *Joseph Lyons* belongs to the Australian Prime Minster or not. Previous work has proposed various approaches to represent words on the space such as latent semantic analysis (LSA) (Deerwester et al., 1990) or Latent Dirichlet Allocation (LDA) (Blei et al., 2003). In particular, the vector-space word representations learned by a neural network have been shown to successfully improve various NLP tasks (Collobert and Weston, 2008; Socher et al., 2013; Nguyen et al., 2015). Our work utilises the skip-gram model as implemented in freely available *word2vec*[14], which is a neural network toolkit introduced by Mikolov et al. (2013), to generate word vectors; they show that *word2vec* is competitive with other vector space models in capturing syntactic and semantic regularities in natural language when trained on the same data.

This work focuses on a name *Joseph Lyons* and we extract all news articles containing the name from Trove. For simplicity, we assume that there is only one *Joseph Lyons* for each year and the name is tagged with the publishing year of an article. For instance, *Joseph Lyons* of 1908 and *Joseph Lyons* of 1940 are represented as *joseph_lyons_1908* and *joseph_lyons_1940* in the extracted news articles, respectively. The total number of *Joseph Lyons* is 133 in this yearly representation. We train the *word2vec* skip-gram model on the extracted news articles and all the *Joseph Lyons* tagged with years are encoded to a 300-dimensional continuous word vector via the *word2vec* model.

The 300-dimensional word vectors of *Joseph Lyons* documents are projected into two-dimensional subspace using t-SNE (van der Maaten and Hinton, 2008) and clustered using the k-means clustering algorithm. We use the bayesian information criterion (BIC) to score the clusters for different values of k; the BIC score is maximum for $k = 4$ and so we select this number of clusters for *Joseph Lyons*. Finally we visualise the clusters on the plot based on the timeline as shown in Figure 5. The red line represents the period in office of Prime Minster *Joseph Lyons* and each colour zone on x-axis denotes one cluster in this figure. *Cluster4* is a close match to the true Prime Minister's time in office while *Cluster 3* shows another possible individual in the

[14] `https://code.google.com/p/word2vec/`

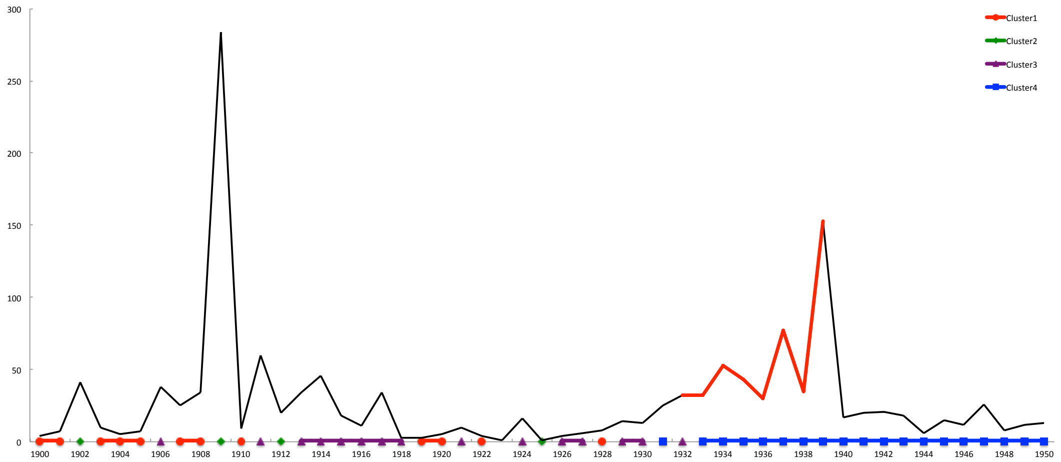


Figure 5: The frequency of mention of the name *Joseph Lyons* with cluster identifiers. The red line represents the period in office of Prime Minster *Joseph Lyons* and each colour zone on x-axis denotes one cluster.

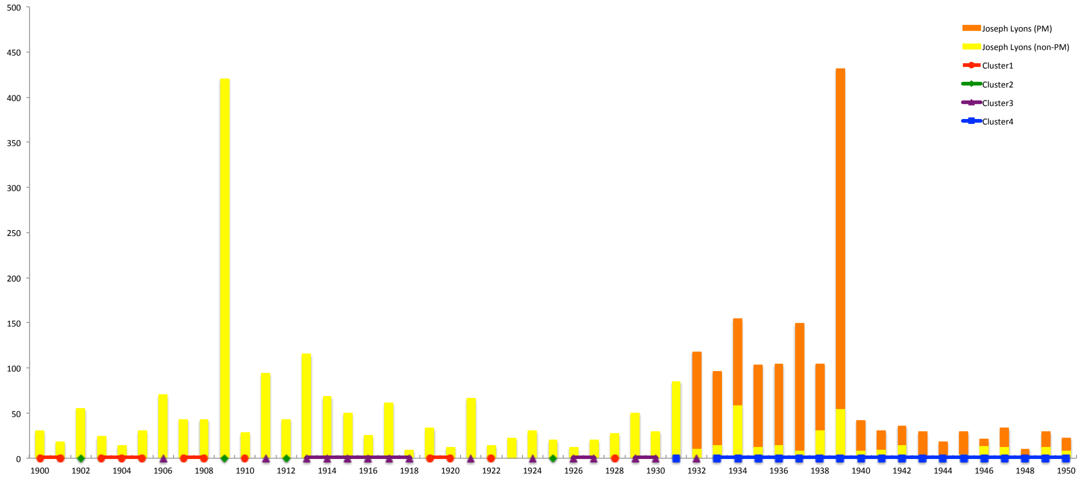


Figure 6: The number of news articles for each year mentioning *Joseph Lyons* as Prime Minster and non Prime Minster along with the clustering results from Figure 5.

period 1913-1918.

To validate the four clusters, we estimate the cluster purity by manually inspecting all news articles containing *Joseph Lyons* and counting those that refer to the PM and those that do not. Figure 6 plots the number of articles for each year mentioning *Joseph Lyons* as PM vs those that are not PM along with the identical clustering results shown in Figure 5. Note that we only count the number of articles of the Prime Minister *Joseph Lyons* and we do not take into account his previous political positions before becoming the Prime Minister.[15] The figure shows that in the region of *Cluster4* the majority of mentions are of the PM while outside this region, the mentions are of a different individual (or *Joseph Lyons* before he was PM). Of the mentions in *Cluster4*, 75% are of the PM.

6 Publishing Linked Data

The results of the NER process have been made available to the HuNI project and will be integrated with their existing data collection as a new data feed linking names with Trove articles. However, we were interested in making a version of this data available in a way that would facilitate further experimentation and exploitation. To this end we have published a version of the data set on the web as *linked data*.

[15] *Joseph Lyons* successively held various Government positions before becoming the tenth Prime Minister of Australia. For instance, he became a new Treasurer of Australia in 1914.

```
<http://trove.nla.gov.au/ndp/del/article/60433109> a cc:Work ;
    dcterms:created "1919-01-10" ;
    dcterms:source <http://trove.alveo.edu.au/source/c987de65b64f0dab35715332478edccd> ;
    dcterms:title "Fatal Accident." ;
    schema:mentions <http://trove.alveo.edu.au/name/7e3030158f7e68d0e161feffd505ee60> ;
    trovenames:context "...hen Young, youngest son of Mr John Young, had the misfortune to meet w
    trovenames:year 1919 .

<http://trove.alveo.edu.au/name/7e3030158f7e68d0e161feffd505ee60> a trovenames:Name ;
    trovenames:word "john",
        "young" ;
    foaf:family_name "young" ;
    foaf:name "John Young" .
```

Figure 7: An example of a named entity mention converted to RDF in turtle format.

The principles of linked data (Berners-Lee et al., 2009) suggest that entities in the data set should be referenced by a URL and that this URL should resolve to a machine readable description of the entity that itself contains URL references to linked entities. A common underlying representation for linked data is RDF. To publish this data set we converted the named entity results to RDF using established vocabularies where possible. This version of the data is then hosted in an RDF triple store and a simple web application has been written to expose the data on the web.

An example of the RDF version of the data is shown in Figure 7. Trove articles are members of the class `cc:Work` and have properties describing publication date, title etc. Each article has one or more `schema:mentions` where each mention is an entity referring to a name. To facilitate searching, each name entity has properties containing the lowercase words in the name as well as the family name and the full name.

The resulting data set consists of 143 million triples and takes up around 26G of database storage using the 4store triple store[16]. A lightweight wrapper was written on this data to provide a web interface to the data set such that all of the URLs in the data resolve to the results of queries and return details of the particular resource. Using HTTP content negotiation, the response will be an HTML page for a web browser or a JSON representation for a script that sends an Accept header of `application/json`.

The API and the web application provide a SPARQL endpoint that supports queries over the data set. The web application is able to visualise the results of queries using the YASGUI[17] query front end.

As an example of mining the named entity data for more information, we wrote queries to find the *associates* of a given name. An associate is a name mentioned in the same document as another name. The query ranks the associated names by frequency of occurrence and returns the top 50 names. So, for example, the associates of *Robert Menzies* can be found at `http://trove.alveo.edu.au/associates/d857a2677bcb9955e286aafe53f61506` which shows that the top five are also politicians:

- Harold Holt (2552)
- Malcolm Fraser (1974)
- John Gorton (1596)
- Paul Hasluck (1232)
- John Curtin (1210)

This simple query shows some of the power that comes from having an accessible data source extracted from the Trove text. In the future we hope to be able to provide more kinds of query and visualisation that will enhance this data source for Humanities researchers.

7 Conclusion and Future Work

This paper has described a project to add value to a Humanities data set using standard NLP systems. The data set itself is interesting as a large collection of historical Australian newspaper text and will be made available via the Alveo virtual laboratory. Using a standard NER process we extracted 27 million person name mentions referencing 17 million articles in the archive. We have shown how this data can be exploited in a number of ways, namely by using a clustering method

[16] http://4store.org/

[17] http://about.yasgui.org/

to try to identify individuals in the data and by presenting the data set as linked data over the web.

The availability of this analysis has already proved interesting to Humanities researchers and we hope to be able to feed it back to the original Trove system run by the National Library of Australia. By providing this as an open data set the NLA encourage collaboration on the data and we hope to do the same with this new named entity data set.

References

Tim Berners-Lee, Christian Bizer, and Tom Heath. 2009. Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Julian Brooke, Adam Hammond, and Graeme Hirst. 2015. Gutentag: an nlp-driven tool for digital humanities research in the project gutenberg corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 42–47, Denver, Colorado, USA, June. Association for Computational Linguistics.

Steve Cassidy, Dominique Estival, Tim Jones, Peter Sefton, Denis Burnham, Jared Burghold, et al. 2014. The Alveo Virtual Laboratory: A web based repository API. In *Proceedings of LREC 2014*, Reykjavik, Iceland.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Rose Holley. 2010. Trove: Innovation in access to information in Australia. *Ariadne*, 64.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Olga Scrivner and Sandra Kübler. 2015. Tools for digital humanities: Enabling access to the old occitan romance of flamenca. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 1–11, Denver, Colorado, USA, June. Association for Computational Linguistics.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August. Association for Computational Linguistics.

Andrew J Torget, Rada Mihalcea, Jon Christensen, and Geoff McGhee. 2011. Mapping texts: Combining text-mining and geo-visualization to unlock the research potential of historical newspapers. *University of North Texas Digital Library*.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Marieke Willems and Rossitza Atanassova. 2015. Europeana newspapers: searching digitized historical newspapers from 23 european countries. *Insights*, 1(28):51–56.

Clinical Information Extraction Using Word Representations

Shervin Malmasi ♣ **Hamed Hassanzadeh** ♢ **Mark Dras** ♣

♣ Centre for Language Technology, Macquarie University, Sydney, NSW, Australia
♢ School of ITEE, The University of Queensland, Brisbane, QLD, Australia

shervin.malmasi@mq.edu.au, h.hassanzadeh@uq.edu.au
mark.dras@mq.edu.au

Abstract

A central task in clinical information extraction is the classification of sentences to identify key information in publications, such as intervention and outcomes. Surface tokens and part-of-speech tags have been the most commonly used feature types for this task. In this paper we evaluate the use of word representations, induced from approximately 100m tokens of unlabelled in-domain data, as a form of semi-supervised learning for this task. We take an approach based on unsupervised word clusters, using the Brown clustering algorithm, with results showing that this method outperforms the standard features. We inspect the induced word representations and the resulting discriminative model features to gain further insights about this approach.

1 Introduction

Evidence-based Medicine (EBM) is an approach to enhance clinical decision making by leveraging currently available evidence. The rationale behind EBM is that clinicians can make more judicious decisions with access to abundant clinical evidence about a particular medical case. This evidence is sourced from research outcomes which can be found in medical publications accessible via online repositories such as PubMed.[1] Although millions of publications are available, finding the most relevant ones is cumbersome using current search technology. Additionally, the rapid growth of research output makes manual analysis and synthesis of search results unfeasible. This has given rise to the need for methods to automatically extract relevant information from publications to support automatic summarization (Hassanzadeh et al., 2015). This is an emerging research area that has begun to attract increasing attention (Summerscales et al., 2011).

This information extraction is generally performed at the sentence level on the paper abstracts (Verbeke et al., 2012). Scholarly publications usually follow a common rhetorical structure that first defines the problem and research aims by introducing background information. They then describe the methodology and finally the outcomes of the research are presented. Abstracts, as the summary of the reported research, generally have the same structure. This information, which can be considered as *scientific artefacts*, can usually be found in the form of whole sentences within the abstracts. More specifically, the artefacts in the clinical research domain have been categorized as Intervention, Population or Problem, Comparison, and Outcome. This is known as the *PICO* scheme (Richardson et al., 1995). Another proposed approach to formalise the rhetorical structure of medical abstracts is the PIBOSO model (Kim et al., 2011), a refined version of the PICO criteria. It contains six classes, rather than four: (i) POPULATION: the group of individuals participating in a study; (ii) INTERVENTION: the act of interfering with a condition to modify it or with a process to change its course; (iii) BACKGROUND: material that places the current study in perspective, *e.g.* work that preceded the current study; information about disease prevalence; etc.; (iv) OUTCOME: a summarisation of the consequences of an intervention; (v) STUDY DESIGN: the type of study that is being described; and (vi) OTHER: other information in the publication.

By comparing these artefacts across publications clinicians can track the evolution of treatments and empirical evidence, allowing them to employ it in their decision making. However, finding and identifying these artefacts is a barrier. To facilitate this process, various approaches have

[1] http://www.ncbi.nlm.nih.gov/pubmed

Shervin Malmasi, Hamed Hassanzadeh and Mark Dras. 2015. Clinical Information Extraction Using Word Representations . In *Proceedings of Australasian Language Technology Association Workshop*, pages 66−74.

been devised to automatically recognise these scientific artefacts in publications (Hassanzadeh et al., 2014a). The most common approach, as discussed in §2, is the use of supervised learning to classify sentences into the various categories.

Separately, another recent trend in Natural Language Processing (NLP) has been the use of word representations to integrate large amounts of unlabelled data into such supervised tasks, a form of semi-supervised learning (Turian et al., 2010). This is something that has not been applied to scientific artefacts extraction.

Accordingly, the primary aim of the present work is to draw together the two areas, evaluating the utility of word representations for this task and comparing them against the most commonly used features to see if they can enhance accuracy. A secondary goal is to inspect the induced word representations and the resulting discriminative models to gain further insights about this approach.

The paper is structured as follows. We present related work on biomedical information extraction in §2. Word representations are introduced in §3 along with our unlabelled data and clustering method. The experimental setup is outlined in §4 followed by results in §5. In §6 we analyze the most discriminative features of our model and in §7 we present a brief error analysis. Finally, we conclude with a discussion in §8.

2 Related Work

The approaches for classifying scientific artefacts vary from having very coarse grained models of these artefacts, such as, publication zone/section identification (Teufel, 2000), to more fine grained ones, such as, sentence classification (Kim et al., 2011; Liakata et al., 2012). In this section, we review the literature that has a similar perspective as ours, that is, sentence-level classification.

Kim et al. (2011) perform classification in two steps using PIBOSO scheme. In the first step, a classifier identifies the sentences that contain PIBOSO concepts, while in the second step, a different classifier assigns PIBOSO classes to these sentences. The annotation is performed at the sentence level and one sentence may have more than one class (*i.e.* multi-label classification). They also employ a Conditional Random Field (CRF) as their classifier using features derived from the context, semantic relations, structure and the sequence of sentences in the text. Domain-specific information is obtained via Metamap. Their final feature vector includes a combination of: bag-of-words, bigrams, part-of-speech (POS) tags, semantic information, section headings, sentence position, and windowed features of the previous sentences.

Verbeke et al. (Verbeke et al., 2012), on the other hand, apply a statistical relational learning approach using a kernel-based learning (kLog) framework to perform classification using the NICTA-PIBOSO corpus. They exploit the relational and background knowledge in abstracts, but take into account only the sequential information at word level. More concretely, their feature set includes a sequence of class labels of the four previous sentences as well as of the two following ones, the lemma of the dependency root of the current sentence and the previous sentence, the position of the sentence, and the section information.

Finally, Sarker et al. (2013) use a set of binary Support Vector Machine (SVM) classifiers in conjunction with feature sets customised for each classification task to attain the same goal. Using the same NICTA-PIBOSO corpus, they use MetaMap to extract medical concepts, and in particular UMLS Concept Unique Identifiers (CUIs) and Semantic Types, to be then considered as domain-specific semantic features. The rest of the features they employ consist of n-grams, POS tags, section headings, relative and absolute sentence positions and sequential features adapted from Kim et al. (2011), as well as class-specific features for the POPULATION class. Similar to our approach, they use an SVM classifier.

A key commonality of previous research is that lexical features and POS tags constitute a set of core features that are almost always used for this task. Although some approaches have applied different external resources, from generic dictionaries such as WordNet to domain specific ontologies, no attempt has been made to leverage large-scale unlabelled data. The main aim of this work is to evaluate the feasibility of such an approach.

3 Word Representations

Word representations are mathematical objects associated with words. This representation is often, but not always, a vector where each dimension is a *word feature* (Turian et al., 2010). Various methods for inducing word representations have been proposed. These include *distributional* representations, such as LSA, LSI and LDA, as well as *distributed* representations, also known as *word embeddings*. Yet another type of representation is based on inducing a clustering over words, with Brown clustering (Brown et al., 1992) being the most well known method. This is the approach that we take in the present study.

Recent work has demonstrated that unsupervised word representations induced from large unlabelled data can be used to improve supervised tasks, a type of semi-supervised learning. Examples of tasks where this has been applied include dependency parsing (Koo et al., 2008), Named Entity Recognition (NER) (Miller et al., 2004), sentiment analysis (Maas et al., 2011) and chunking (Turian et al., 2010). Such an approach could also be applied to the clinical information extraction task where although we only have a very limited amount of labelled data, large-scale unlabelled data — hundreds of millions of tokens — is readily available to us.

Researchers have noted a number of advantages to using word representations in supervised learning tasks. They produce substantially more compact models compared to fully *lexicalized* approaches where feature vectors have the same length as the entire vocabulary and suffer from sparsity. They better estimate the values for words that are rare or unseen in the training data. During testing, they can handle words that do not appear in the labelled training data but are observed in the test data and unlabelled data used to induce word representations. Finally, once induced, word representations are model-agnostic and can be shared between researchers and easily incorporated into an existing supervised learning system.

3.1 Brown Clustering

We use the Brown clustering algorithm (Brown et al., 1992) to induce our word representations. This method partitions words into a set of c classes which are arranged hierarchically. This is done through greedy agglomerative merges which optimize the likelihood of a hidden Markov model which assigns each lexical type to a single class. Brown clusters have been successfully used in tasks such as POS tagging (Owoputi et al., 2013) and chunking (Turian et al., 2010). They have been successfully applied in supervised learning tasks (Miller et al., 2004) and thus we also adopt their use here.

3.2 Unlabelled Data

To obtain suitable unlabelled data, we followed two strategies to retrieve data from the PubMed repository: *(1)* based on user-defined clinical inquiries, and *(2)* using a generic query. In the first strategy we employed 456 clinical queries from the EBMSummariser corpus (Mollá and Santiago-martinez, 2011). The inquiries in this corpus are collected from the Clinical Inquiries section of the Journal of Family Practice.[2] This section of the journal contains a number of queries submitted by the users and their evidence-based answers by medical experts. We queried PubMed with these 456 inquiries and retrieved the results using their PM-IDs (*i.e.* PubMed's unique identifiers) via PubMed's eUtils API.[3] In total, 212,393 abstracts were retrieved, of which 22,873 abstracts did not contain valid text, leaving 189,520.

For the second retrieval strategy, we queried PubMed with the term *Randomised Controlled Trial*. This results in retrieving publications presenting medical cases and providing evidence (*i.e.* desirable for EBM practice). PubMed returned 491,357 results for this query. After removing duplicate results, *i.e.* those retrieved in the first strategy, we downloaded 200,000 abstracts. After removing empty abstracts, 171,662 remained.

The text of each abstract was extracted by parsing the PubMed XML file and it was then segmented into sentences; each sentence was then tokenized and lowercased. This resulted in a total of 96 million tokens across 3.7 million sentences, with 873k unique tokens.[4]

We next induced Brown clusters using this data. Five runs with clusters of size 100, 200, 300, 1000 and 3000 were performed for comparison purposes.

3.3 Clustering Results

We now turn to a brief analysis of the clustering results. Table 1 shows examples of both generic and domain-specific clusters taken from the run with 3,000 clusters. We observe that words were clustered according to both their semantic and grammatical properties, with some clusters containing highly domain-specific entries. These results show that the word clusters are very effective at captur-

[2] http://jfponline.com/articles/clinical-inquiries.html

[3] http://www.ncbi.nlm.nih.gov/books/NBK25497/

[4] We also note that this data has a much higher type-token ratio compared to other domains such as newswire text, indicating greater lexical variation in this domain.

Cluster Path	Top Words
00100111	article paper manuscript chapter commentary essay
001011011010	observations investigations evidences facts explorations
1000000011	evaluating investigating examining exploring
111010111011100	suggests indicates implies posits asserts contends
111010111011101	shows demonstrates reveals confirms concludes argues establishes assumes finds
1111011011001	mg/dl mmhg kg/m2 bpm beats/min u/ml mmol/mol
001111000101	antibiotics analgesics opioids antimicrobials placebos antihypertensives
11000100100	reconstruction dissection ligation instrumentation
010111101011110	oncology cardiology rheumatology psychiatry urology dermatology radiology
010111100011010	vaccination immunization inoculation immunisation immunizations revaccination

Table 1: Some example clusters and their top words (by frequency). Examples include both generic (top) and domain-specific (bottom) clusters.

ing lexical knowledge and organizing it by syntactic function. We will examine the cluster contents again in §6 as part of our feature analysis. We make these unsupervised clusters available for viewing or download from our website.[5]

4 Experimental Setup

We take a supervised classification approach, comparing previously used features against the unsupervised Brown cluster features.

As the primary focus of this work is the evaluation of word representation features, we limit the scope of our experiment in two ways: (1) we do not attempt multi-label classification, as explained in §4.1 and (2) we do not use sentence sequence information, as outlined in §4.2. These conditions allow us to focus on systematically comparing feature types in a controlled manner.

4.1 Data

We use the NICTA-PIBOSO corpus (Kim et al., 2011) in this experiment. Here each sentence is labelled with one or more classes, making it a multi-label classification task. Table 2 lists a breakdown of the per-class sentence statistics, showing that 9% of the sentences have more than one label. The multi-label characteristic of instances as well as imbalanced distribution of classes are two most common issues of many corpora in biomedical scientific artefacts classification task (Hassanzadeh et al., 2014b). As the scope of our work is limited to evaluating word representation features, we simplify our setup by excluding the multi-label instances, thus reducing the task to a multi-class classification one. This avoids the use of multi-label evaluation metrics, making it easier to draw direct comparisons between the performance of the standard features and the word representations. The sentences were tokenized in a preprocessing step.

	All	Multi-label
BACKGROUND	2,557	160 (6%)
INTERVENTION	690	350 (51%)
OUTCOME	4,523	71 (2%)
POPULATION	812	412 (51%)
STUDY DESIGN	228	114 (50%)
OTHER	3,396	0 (0%)
Total	12,206	1,107 (9%)

Table 2: Sentence counts in the NICTA-PIBOSO corpus. The multi-label column lists the number of sentences annotated with more than one label.

4.2 Classifier

We use a linear SVM to perform multi-class classification. In particular, we use the LIBLINEAR[6] package (Fan et al., 2008) which has been shown to be efficient for highly-dimensional text classification problems such as this (Malmasi and Dras, 2014; Malmasi and Dras, 2015b; Malmasi and Dras, 2015a).

Previous work (see §2) shows that CRF classifiers perform well for this task, exploiting the sequential structure of abstracts. As our aim is to evaluate the effectiveness of *intrinsic* word representation features we focus on the classification of individual sentences and do not use *extrinsic* features, *i.e.* the contents or predicted labels of preceding sentences in an abstract. In practice this means that the sentences are being classified independently.

[5] http://web.science.mq.edu.au/%7Esmalmasi/data/med3k/

[6] http://www.csie.ntu.edu.tw/%7Ecjlin/liblinear/

4.3 Features

We compare our proposed word representation features against the most commonly used features for this task, which we describe here.

Word n-grams Surface tokens are the most commonly employed feature type in this task using both bag-of-words (unigram) and n-grams. The length of the feature vector equals that of the vocabulary; n-gram vocabulary grows exponentially. We extracted word n-grams of order 1–3.

Part-of-Speech n-grams POS tags are another frequently used feature type and capture the syntactic differences between the different classes.[7] We tagged the sentences using the Stanford Tagger, which uses the Penn Treebank tagset containing 36 tags, and extracted n-grams of order 1–3.

Brown Cluster Features Brown clusters are arranged hierarchically in a binary tree where each cluster is identified by a bitstring of length ≤ 16 that represents its unique tree path. The bitstring associated with each word can be used as a feature in discriminative models, Additionally, previous work often also uses a p-length prefix of this bitstring as a feature. When p is smaller than the bitstring's length, the prefix represents an ancestor node in the binary tree and this superset includes all words below that node. We follow the same approach here, using all prefix lengths $p \in \{2, 4, 6, \dots, 16\}$. Using the prefix features in this way enables the use of cluster supersets as features and has been found to be effective in other tasks (Owoputi et al., 2013). Each word in a sentence is assigned to a Brown cluster and the features are extracted from this cluster's bitstring.

4.4 Evaluation

We report our results as classification accuracy under k-fold cross-validation, with $k = 10$. These results are compared against a majority baseline and an oracle. The oracle considers the predictions by all the classifiers in Table 3 and will assign the correct class label for an instance if at least one of the the classifiers produces the correct label for that data point. This approach can help us quantify the *potential* upper limit of a classification system's performance on the given data and features (Malmasi et al., 2015).

Feature	Accuracy (%)
Majority Baseline	40.1
Oracle	92.5
Part-of-Speech unigrams	64.6
Part-of-Speech bigrams	68.6
Part-of-Speech trigrams	67.4
Word unigrams	73.3
Word bigrams	66.0
Word trigrams	49.7
Brown (100 clusters)	70.4
Brown (200 clusters)	72.8
Brown (300 clusters)	74.3
Brown (1000 clusters)	74.8
Brown (1000 clusters) bigrams	73.9
Brown (3000 clusters)	**75.6**
Brown (3000 clusters) bigrams	74.9
Brown (3000 clusters) trigrams	70.7

Table 3: Sentence classification accuracy results for the features used in this study.

5 Results

The results for all of our experiments are listed in Table 3. All features performed substantially higher than the baseline. We first tested the POS n-gram features, with bigrams providing the best result of 68.6% accuracy and performance dropping with trigrams. Word n-grams were tested next, with unigrams achieving the best result of 73.3%. Unlike the POS features, word feature performance does not increase with bigrams.

Finally, the Brown cluster features were tested using clusters induced from the five runs of different cluster different sizes. Accuracy increases with the number of clusters; 200 clusters match the performance of the raw unigram features and the largest cluster of size 3000 yields the best result of 75.6%, coming within 17% of the oracle accuracy of 92.5%. Another variation tested was Brown cluster n-grams. Although they outperformed their word n-gram counterparts, they did not provide any improvement over the standard Brown features.

In sum, these results show that Brown clusters, using far fewer features, can outperform the widely used word features.

[7] *e.g.* Our own analysis showed that OUTCOME sentences contained substantially more past tense verbs, comparative adverbs and comparative adjectives.

Class	Clusters of words
BACKGROUND	[have has had] — [describes presents examines discusses summarizes addresses] [objectives goal] — [emerged evolved attracted fallen arisen risen proliferated]
INTERVENTION	[received underwent undergoing taking] — [gel cream spray ointment] [orally intravenously subcutaneously intramuscularly topically intraperitoneally] [mg/kg mg/kg/day g/kg ml/kg μg/kg mg/kg/d microg/kg μg/kg]
POPULATION	[identified enrolled recruited contacted] — [aged] — [randomly] [twenty thirty forty sixty fifty eighty thirty-two twenty-eight . . .]
OUTCOME	[revealed showed suggests indicates implies] — [found observed noted noticed] [significantly] — [p n r r2] — [demonstrate indicate imply] [0.002 0.003 0.004 0.006 .02 0.008 0.007 .03 0.009 .04 . . .]
STUDY DESIGN	[cross-sectional case-control quasi-experimental sectional mixed-methods case-crossover case-controlled . . .] — [randomised randomized-controlled]
OTHER	[include] — [evaluate assess] — [obtained] — [measured] [articles papers publications literatures manuscripts]

Table 4: Some highly-weighted clusters associated with the NICTA-PIBOSO classes. Each cluster is a single feature in the model, but we have expanded them here to include their constituent words.

6 Feature Analysis

In this section we analyze some of the discriminative features in our model to gain better insight about the knowledge being captured by our models and the task in general. This was done by ranking the features according to the weights assigned by the SVM model. In this manner, SVMs have been successfully applied in data mining and knowledge discovery tasks such as identifying discriminant cancer genes (Guyon et al., 2002).

Table 4 lists several highly weighted Brown clusters for each of our classes. Although each cluster is a single feature in the model, we have expanded the clusters here to include their constituent words.

The BACKGROUND class is associated with words that are quite common in the introductory rhetoric of scientific publications. These are descriptive of the current and previous research, and are mostly in the present/past perfect tense.

The INTERVENTION class is mostly associated with clusters that include clinical vocabulary, including verbs such as *received*, *underwent* and *taking*; medication-related nouns like *gel* or *ointment*; dosage descriptors such as *mg/kg* and *mg/kg/day*; and adverbs describing the route of administration, for example *orally* and *intravenously*.

For POPULATION sentences, numerical quantities, likely relating to the number of participants,[8] as well as verbs that are related to participation, are very frequent.

Similarly, reporting verbs are more likely to occur in OUTCOME sentences. They are organized into different clusters according to their syntactic and semantic function. In addition, we also note that a cluster of decimal numbers is also common. These numbers are used in the sentences to report study results, including those from various statistical tests. This is accompanied by another cluster containing relevant tokens for reporting statistics, *e.g.* “p”, “r”, and “n” which could refer to “p-value”, “Pearson correlation” and “number”, respectively.

Overall, it can be seen that the clusters associated with the features are logical. Furthermore, these examples underline the clustering method’s effectiveness, enabling us to encode a wide range of similar tokens (*e.g.* decimal values or dosage

[8] These are mostly spelled out as they appear at the start of a sentence.

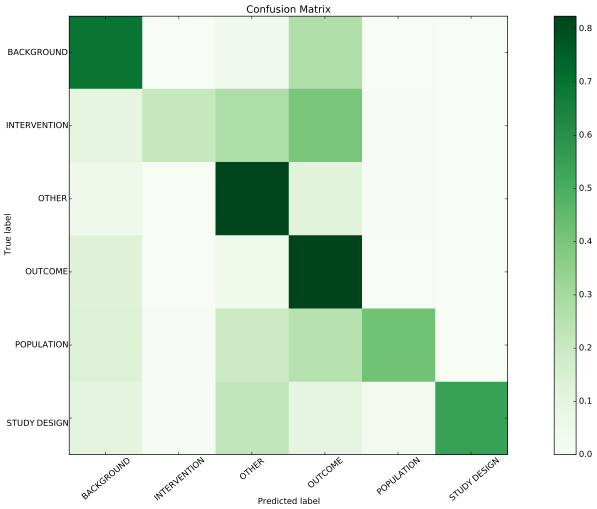


Figure 1: Normalized confusion matrix for results using Brown features (3000 clusters). The values are normalized due to the class size imbalance.

amounts) under a single cluster feature. This provides a substantial reduction in the feature space without the loss of information.

7 Error Analysis

We now turn to an analysis of the errors being committed by the classifier. The error distribution is illustrated by the confusion matrix in Figure 1. We note that the two largest classes, OUTCOME and OTHER, are the most correctly classified. Conversely, INTERVENTION sentences are highly misclassified and mostly confused for OUTCOME. To better understand these errors we segregated the subset of misclassified instances for analysis. Table 5 lists a number of these sentences from highly confused classes.

Our analysis suggests that the occurrences of similar domain-specific terminologies in both types of sentences, in INTERVENTION sentences as the explanation of the methodologies, and restating them in OUTCOME sentences in order to describe the effects of those methodologies, can be a reason for this confusion.

There is also some confusion between BACKGROUND and OUTCOME instances. Both of these classes commonly describe some challenges and findings of either previous studies (*i.e.* BACKGROUND sentences) or the current reporting study (*i.e.* OUTCOME). This narrative characteristic of these classes has similar rhetorical and linguistic attributes, *e.g.* they usually contain past tense verbs and similar structures. This is demonstrated by the two example OUTCOME sentences in Table 5 which are misclassified. Looking at the sentences, it can be challenging even for a human to correctly label them without knowing the context; they both describe the outcome of a study, but it is not clear if it is the reporting study or previous work. Only by reading it in the context of the abstract and the preceding sentence can we confidently determine that they are outcomes of the present study. This is the case for many of the misclassified instances.

However, this is not due to the feature types but rather the classification approach taken here and in many other studies for this task. The SVM does not model the sequential characteristics of sentences in an abstract, instead classifying them independently. It is mostly for these reasons that sequence labelling algorithms, *e.g.* Conditional Random Fields (CRF), have been found to be useful for this task, as we mentioned in §2. Hence, it has been noted that applying such methods with the most suitable features can considerably avoid such contextual errors and improve the overall accuracy (Jonnalagadda et al., 2015).

8 Discussion

We presented a semi-supervised classification approach for clinical information extraction based on unsupervised word representations, outperforming the most commonly used feature types. This is the first application of word representation features for this task; the promising results here inform current research by introducing a new feature class. We also made our word clusters available.

A positive byproduct of this approach is a substantial reduction in the feature space, and thus model sparsity. This has practical implications, resulting in more efficient models and enabling the use of simpler learning algorithms which are generally used with smaller feature sets. This would allow faster and more efficient processing of large amount of data which is an important practical facet of this task. For example, we conducted some preliminary experiments with multinomial Naïve Bayes and k-NN classifiers and our results showed that the Brown cluster features achieved faster and much more accurate results than a bag-of-words approach.

Actual	Predicted	Sentence
INTERVENTION	OUTCOME	Glucocorticoids were decreased and could be stopped as the neurologic deficits fully recovered.
INTERVENTION	OTHER	Subjects were examined before and 1 year after surgical treatment.
OUTCOME	BACKGROUND	Negative symptoms are associated with poor outcome, cognitive impairments, and incapacity in social and work domains.
OUTCOME	BACKGROUND	Patients suffering from mild TBI are characterized by subtle neurocognitive deficits in the weeks directly following the trauma.
POPULATION	OTHER	The aim of this study was to investigate this association in an Italian OCD study group.
POPULATION	OUTCOME	Five cases of biopsy- or Kveim test-proved sarcoidosis with MR findings consistent with MS are reported.

Table 5: Examples of misclassified sentences with their true and predicted labels.

One limitation here was the size of the unlabelled data we used for inducing the Brown clusters.[9] Future work could examine the effects of using more data on classification accuracy.

Having demonstrated the utility of the features, there are a number of directions for future work. We previously described that sequence labelling approaches have been found to be helpful for this task given the structured nature of the abstracts. At the same time, it has been shown that incorporating word representations can result in significant improvements for sequence labelling tasks (Huang and Yates, 2009; Turian et al., 2010; Miller et al., 2004). Therefore, the combination of these two approaches for this task seems like a natural extension.

The evaluation of these Brown cluster features on other datasets used for this task — such as the ART corpus (Liakata et al., 2012) — is another direction for research in order to assess if these results and patterns can be replicated.

Cross-corpus studies have been conducted for various data-driven NLP tasks, including parsing (Gildea, 2001), Word Sense Disambiguation (WSD) (Escudero et al., 2000) and NER (Nothman et al., 2009). While most such experiments show a drop in performance, the effect varies widely across tasks, making it hard to predict the expected drop. This is something that could be evaluated for this task by future work.

Finally, previous work has also found that combining different word representations can further improve accuracy, *e.g.* the results from Turian et al. (2010, §7.4). This is another avenue for further research in this area.

References

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

Gerard Escudero, Lluís Màrquez, and German Rigau. 2000. An empirical study of the domain dependence of supervised word sense disambiguation systems. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 172–180.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Daniel Gildea. 2001. Corpus variation and parser performance. In *EMNLP*, pages 167–202.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

Hamed Hassanzadeh, Tudor Groza, and Jane Hunter. 2014a. Identifying scientific artefacts in biomedical literature: The evidence based medicine use case. *Journal of Biomedical Informatics*, 49:159 – 170.

Hamed Hassanzadeh, Tudor Groza, Anthony Nguyen, and Jane Hunter. 2014b. Load balancing for imbalanced data sets: Classifying scientific artefacts for evidence based medicine. In *PRICAI 2014: Trends in Artificial Intelligence*, volume 8862 of *Lecture Notes in Computer Science*, pages 972–984.

[9] *e.g.* Owoputi et al. (2013) used approx 850m tokens of unlabelled text compared to our 96m.

Hamed Hassanzadeh, Diego Mollá, Tudor Groza, Anthony Nguyen, and Jane Hunter. 2015. Similarity Metrics for Clustering PubMed Abstracts for Evidence Based Medicine. In *Proceedings of the Australasian Language Technology Workshop (ALTA)*.

Fei Huang and Alexander Yates. 2009. Distributional representations for handling sparsity in supervised sequence-labeling. In *ACL*, pages 495–503. Association for Computational Linguistics.

Siddhartha R. Jonnalagadda, Pawan Goyal, and Mark D. Huffman. 2015. New directions in biomedical text annotations: definitions, guidelines and corpus construction. *Systematic Reviews*, 4(78):16.

Su Nam Kim, David Martinez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support Evidence Based Medicine. *BMC Bioinformatics*, 12(Suppl 2):S5.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *ACL*, pages 595–603.

Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. 2012. Automatic recognition of conceptualisation zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.

Shervin Malmasi and Mark Dras. 2014. Chinese Native Language Identification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL-14)*, pages 95–99, Gothenburg, Sweden, April. Association for Computational Linguistics.

Shervin Malmasi and Mark Dras. 2015a. Large-scale Native Language Identification with Cross-Corpus Evaluation. In *NAACL*, pages 1403–1409, Denver, CO, USA, June.

Shervin Malmasi and Mark Dras. 2015b. Multilingual Native Language Identification. In *Natural Language Engineering*.

Shervin Malmasi, Joel Tetreault, and Mark Dras. 2015. Oracle and Human Baselines for Native Language Identification. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 172–178, Denver, Colorado, June.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *HLT-NAACL*, pages 337–342.

Diego Mollá and Maria Elena Santiago-martinez. 2011. Development of a corpus for evidence based medicine summarisation. In *Proceedings of the Australasian Language Technology Association Workshop*.

Joel Nothman, Tara Murphy, and James R Curran. 2009. Analysing Wikipedia and gold-standard corpora for NER training. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620. Association for Computational Linguistics.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *NAACL*. Association for Computational Linguistics.

W.S. Richardson, M.C. Wilson, J. Nishikawa, and R.S. Hayward. 1995. The well-built clinical question: a key to evidence-based decisions. *ACP Journal Club*, 123(3):A12–A13.

Abeed Sarker, Diego Molla, and Cecile Paris. 2013. An Approach for Automatic Multi-label Classification of Medical Sentences. In *Proceedings of the 4th International Louhi Workshop on Health Document Text Mining and Information Analysis*, Sydney, NSW, Australia.

Rodney L Summerscales, Shlomo Argamon, Shangda Bai, Jordan Huperff, and Alan Schwartz. 2011. Automatic summarization of results from clinical trials. In *Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on*, pages 372–377. IEEE.

Simone Teufel. 2000. *Argumentative Zoning: Information Extraction from Scientific Text*. Ph.D. thesis, School of Cognitive Science, University of Edinburgh, Edinburgh, UK.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Mathias Verbeke, Vincent Van Asch, Roser Morante, Paolo Frasconi, Walter Daelemans, and Luc De Raedt. 2012. A statistical relational learning approach to identifying evidence based medicine categories. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 579–589, Jeju Island, Korea.

How few is too few? Determining the minimum acceptable number of LSA dimensions to visualise text cohesion with Lex

Caroline McKinnon
School of Communication and Arts
The University of Queensland
Brisbane, Australia
c.mckinnon@uq.edu.au

Ibtehal Baazeem
School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, Australia
ibtehal.baazeem@uq.net.au

National Center for Computer Technology and Applied Mathematics
King Abdulaziz City for Science and Technology
Riyadh, Saudi Arabia
ibaazeem@kacst.edu.sa

Daniel Angus
School of Communication and Arts
The University of Queensland
Brisbane, Australia
d.angus@uq.edu.au

Abstract

Building comprehensive language models using latent semantic analysis (LSA) requires substantial processing power. At the ideal parameters suggested in the literature (for an overview, see Bradford, 2008) it can take up to several hours, or even days, to complete. For linguistic researchers, this extensive processing time is inconvenient but tolerated—but when LSA is deployed in commercial software targeted at non-specialists, these processing times become untenable. One way to reduce processing time is to reduce the number of dimensions used to build the model. While the existing research has found that the model's reliability starts to degrade as dimensions are reduced, the point at which reliability becomes unacceptably poor varies greatly depending on the application. Therefore, in this paper, we set out to determine the lowest number of LSA dimensions that can still produce an acceptably reliable language model for our particular application: Lex, a visual cohesion analysis tool. We found that, across all three texts that we analysed, the cohesion-relevant visual motifs created by Lex start to become apparent and consistent at 50 retained dimensions.

1 Introduction

Latent Semantic Analysis (LSA) is a well-established method for describing the semantic content in textual data as a set of vectors in a high dimensional semantic space (Wade-Stein & Kintsch, 2004). It is used for a range of applications across a range of fields, including linguistics, cognitive science, education, information science and text analysis (Evangelopoulos, Zhang, & Prybutok, 2012), and it has been verified as an effective method in the majority of these fields both practically and theoretically (Evangelopoulos et al., 2012; Wade-Stein & Kintsch, 2004).

2 Lex: an overview

The application of LSA we are focusing on, pioneered by Foltz, Kintsch, & Landauer (1998), is its use in predicting the coherence of a piece of text by identifying and measuring its lexical cohesive ties. Building on this work, we have designed an LSA-based tool, which we have called Lex, to allow writers and editors to visually analyse the cohesion—and, by extension, coherence—of their own text. Users upload their text, and the tool derives the relatedness of meaning that occurs in each sentence throughout the text using a LSA language model by investigating word usage patterns in a large text corpus

Caroline Mckinnon, Ibtehal Baazeem and Daniel Angus. 2015. How few is too few? Determining the minimum acceptable number of LSA dimensions to visualise text cohesion with Lex . In *Proceedings of Australasian Language Technology Association Workshop*, pages 75−83.

(McCarthy, Briner, Rus, & McNamara, 2007), then maps out the strength of the conceptual match between every pair of sentences to a recurrence plot visualisation. The intensity of shading in each block increases with the strength of the match: shading represents more shared concepts and a higher level of cohesion between the two sentences, and paler shading or whitespace represents fewer shared concepts and less cohesion. Users can then use the visualisation to assess the overall cohesion level of their document, quickly locate areas of low cohesion that may need improving, or discover any other cohesion-relevant patterns that would otherwise have been difficult to detect.

Though it has yet to be subjected to thorough empirical testing at this early stage, we theorise that this visualisation-based method should provide a more efficient method of cohesion analysis than the traditional manual approach, because it takes advantage of the high-bandwidth, pre-attentive processing that visual perception enables (Ware, 2013). Especially in larger documents of more than a few pages, an editor's ability to detect cohesion problems is limited by their working memory capacity—by the time they get to the end of the document, they have forgotten what was at the beginning (Kintsch, Patel, & Ericsson, 1999).

In practice, we see Lex as particularly useful for a large organization such as, for example, a Queensland Government department. we most likely see Lex being used by communication staff as part of their editing process. It could help them diagnose potential problems and identify areas requiring editing or restructuring in documents intended for the public, thereby helping to ensure that the documents are cohesive enough to be clearly understood. Government organisations in particular stand to gain from clear communication: studies have shown links with improved public understanding of and increased compliance with regulations, reduced time and resources devoted to answering questions, and even greater support for government and its initiatives (Watson & Lynch, 1998). Especially in the case of guidelines or policies, unclear communication can have ethical and legal consequences, raising the question of whether citizens can be expected to comply with guidelines or laws that they are not able to fully understand (Austen, Gilbert, & Mitchell, 2001).

3 The problem

Using a pre-generated language model to analyse the user's text is not ideal for Lex's purposes. To be most useful, it needs to be able to provide reliable results when analysing any text genre or style (within reason), but the literature clearly establishes that the reliability of a result is affected significantly by the semantic similarity of the corpus text that a language model is generated from. The more similar the corpus is to the text being analysed, the more reliable the results (Biber, 1993; Koester, 2010). One way to get around this problem is to supply a range of readymade language models based on broad genres (fiction, academic, journalistic) but also offer the user the option to supply a corpus of their own that is more similar to the text they wish to analyse, and have the tool build a language model from that in run time. However, building a language model at the specifications that most literature recommends is a resource-intensive, time-consuming computational process, beyond the capability of the average desktop PC (not to mention the average user's patience) (Bradford, 2008).

One impediment is the need to use a very large corpus: the literature often recommends, on the whole, using very large corpora in the vicinity of 10 million words (Landauer, Foltz, & Laham, 1998), which can be extremely resource intensive to process. However, quality, rather than quality, is more important when it comes to corpus size: in other words, the size of the corpus could be reduced significantly without sacrificing too much by way of performance if it is highly semantically similar to the text to be analysed (Biber, 1993; Koester, 2010). The other restriction is the number of dimensions retained in the semantic space—the higher the number of dimensions retained, the more resource-intensive the process (Bradford, 2008). The bulk of studies conducted broadly appear to recommend 300 dimensions as the ideal number for LSA, but individual studies have settled on anywhere between six (Lerman, 1999) and close to 2000 (Efron, 2005). The experiments conducted to arrive at these specifications vary broadly in purpose, and use vastly different corpora types and sizes, though, which explains the large variation in findings. Reducing the number of dimensions required to produce acceptably reliable results for Lex could make this 'custom corpus' option viable, by reducing the processing time to within reasonable limits. We suspected that a highly semantically similar, small corpus would require fewer retained dimensions to perform at acceptable accuracy levels than a large, generalised one—potentially far fewer than the industry standard of 300. What we needed to determine, though, was just how few dimensions we could retain in our semantic space before the analysis results became unreliable.

4 Method

In order to find out where the acceptability threshold lies, we generated eight LSA recurrence plots each for three different samples of text, setting the number of dimensions retained to a different threshold each time (10, 20, 50, 70, 100, 300, 500, and 700)—in total, 24 recurrence plots. We then conducted qualitative visual analyses to identify several lexical-cohesion-relevant patterns—which we will call 'motifs'—that were readily apparent in the 300-dimension versions of the plot. Three hundred was the threshold we chose as the 'gold standard' because, as discussed, it is most often recommended in the literature, and what it showed aligned most closely to our own expert assessment of the cohesion patterns in the text. We then searched for the motifs in the plots generated at successively lower dimensions, aiming to determine the lowest dimension interval at which they were still easily recognisable.

The texts we used for analysis are small sub-sections (of between 700 and 1400 words) of three different Queensland government-affiliated reports: *Delivering continuity of midwifery care to Queensland women: A guide to implementation* (Queensland Government, 2012) (herein known as the "Midwives report"); *A shared challenge: Improving literacy, numeracy, and science skills in Queensland Primary Schools* (Masters, 2009), (the "Education report"); and *Not Now, Not Ever: Putting an end to domestic violence in Queensland* (Special Taskforce on Domestic and Family Violence in Queensland, 2015), (the "Domestic violence report"). The reports in full were all around 45,000 words in length each (before pre-processing), and for each text sample we analysed, we used the full text of the report from which it came as a corpus to generate the language model. Based on our experience, 40 – 50,000 words is likely to be as large a corpus as most non-specialists could conveniently locate, so these reports imitated the conditions under which Lex would likely be used—and, as previously discussed, these corpora may be small compared to what is often recommended for building LSA language models, but what they lack in size, they make up for in specificity.

We chose government reports because government writers and editors are potential target users for technology such as Lex: they regularly produce long, complex documents for audiences with limited domain knowledge, a scenario in which cohesion is known to significantly affect readers' comprehension (McNamara, Kintsch, Songer, & Kintsch, 1996).

To appropriately test this tool for its intended purpose, we deliberately selected naturalistic data—documents that are, on the whole, fairly cohesive to begin with (as opposed to, for example, putting together random groups of sentences to artificially create or exaggerate the presence of motifs). They all certainly meet the minimum threshold to be coherent, so we knew that any detectable motifs were likely to be subtle.

The Lex plots were compared using a mixed-methods approach. Qualitative interpretation was used to determine the presence or absence of macro and meso-scale features (motifs), and a quantitative distance measure was used to summarise the magnitude of difference between the plots. For the quantitative measure all possible pairings of plots from the same test document were calculated. The measure designed for this study was the absolute difference between the plots, expressed as a percentage. The magnitude of the difference between all paired cells was calculated and averaged as:

$$\%\text{dif} = \frac{\sum_{i=0}^{n}\sum_{j=0}^{n}\left|\boldsymbol{M}_{ij} - \boldsymbol{N}_{\boldsymbol{ij}}\right|}{n(n-1)/2} \times 100$$

Where: ***M*** and ***N*** are Lex matrices being compared, and *n* is the total number of plot elements.

5 Results

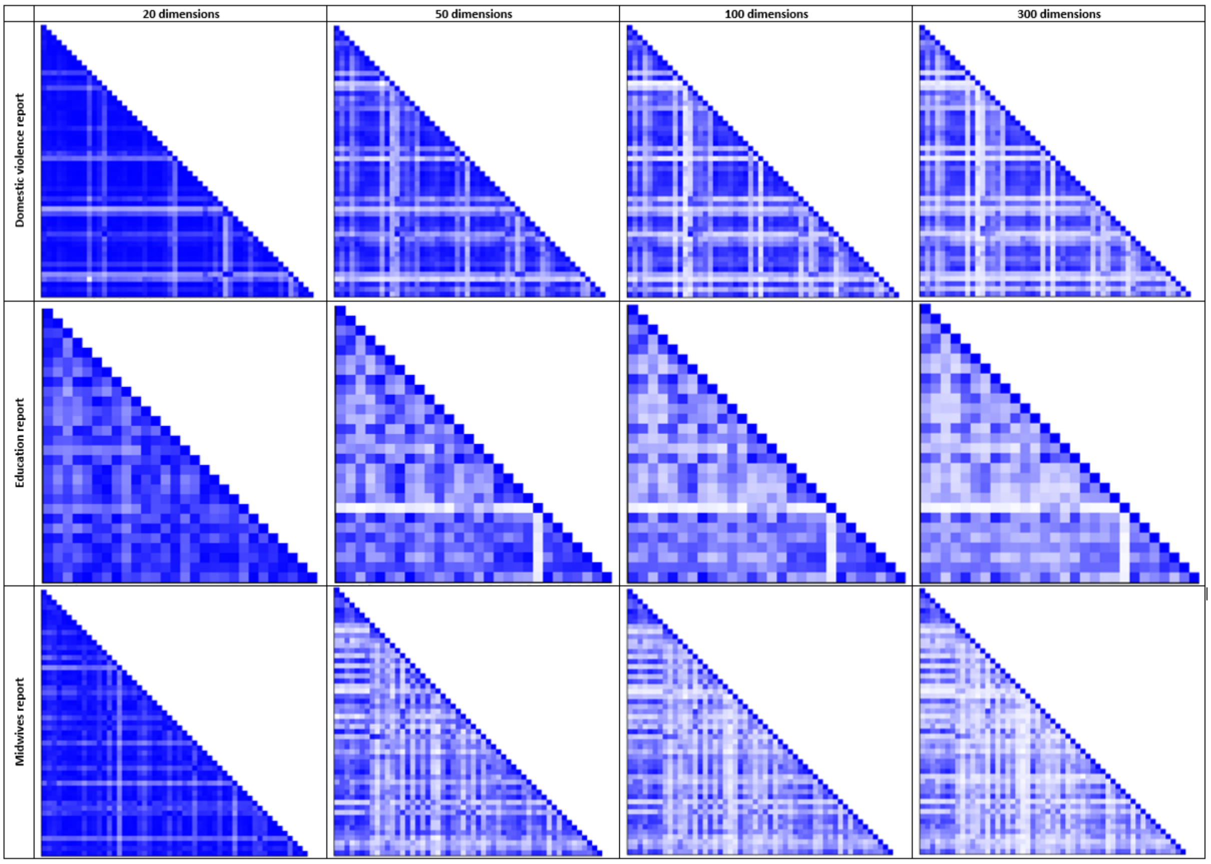


Figure 1. Lex plots for all texts at 20, 50, 100, and 300 dimensions

Domestic Violence:

	20	50	100
50	23%		
100	34%	11%	
300	43%	20%	9%

Education:

	20	50	100
50	20%		
100	28%	9%	
300	36%	16%	8%

Midwives:

	20	50	100
50	36%		
100	45%	9%	
300	53%	17%	8%

Table 1: Absolute difference between the plots

5.1 Reading the Lex plot

Each block along the right diagonal edge of the plot represents a sentence in the document. The document is laid out as a time series, progressing from the first sentence in the top left moving down toward the last sentence in the bottom right. Each variously shaded block in the plot represents the presence (or absence) and strength of the tie between the pair of sentences at whose intersection it sits. The more saturated the shade, the more shared concepts between that pair of sentences. Fainter shading, fewer shared concepts. If no link at all is present, it shows up as white space. In this way, the plot shows the degree of relatedness between every pair of sentences in the document.

At a broad level, a more densely shaded plot can be seen to represent a more globally cohesive document, and a sparse, pale plot represents a less globally cohesive one. But it's the plot's ability to show mesoscale *patterns* of cohesion that are otherwise difficult to detect that separates it from existing methods, such as, for example, the set of cohesion metrics provided by CohMetrix (Graesser, McNamara, Louwerse, & Cai, 2004). The analyses below demonstrate several examples of cohesion-relevant motifs, but only those that happen to be present in the texts we are analysing here: these are by no means an exhaustive set.

5.2 Domestic violence report

At 300 dimensions, the most obvious motifs in the Domestic violence report are the grid-like series of pale stripes criss-crossing the plot at sentences 2, 4-5, 10, 12-17, 25, 27, 35, 37-38, 42-46, 50-51, and 53:

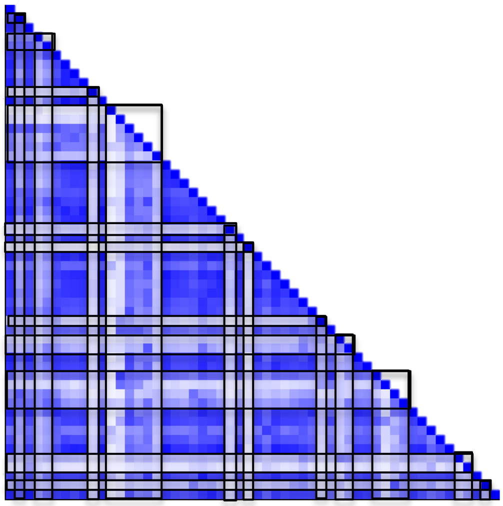


Figure 2: Motifs in 300-dimension Lex plot of Domestic violence report

Although these may present at first glance as problematic low cohesion, on closer inspection, they are actually false alarms—or at least, examples of when lexical cohesion alone cannot always tell the whole cohesion story. Almost all are quite short sentences: for example, sentence 27 reads 'It must not be accepted or excused'. Shorter sentences obviously provide fewer opportunities for content words to occur, which in turn provides fewer opportunities for lexical repetition—though other forms of cohesion may be present, such as the co-reference occurring in sentence 27 with the word 'it'. This highlights a limitation of the method, which we may need to address in future iterations of Lex by normalising for sentence length. Nevertheless, these short sentences are justifiably detected by the algorithm as having little to no semantic similarity to other sentences in the text, and are represented prominently in the visualisation at 300 dimensions, so we have included them in our definition of a motif for the purposes of this exercise.

At ten dimensions, the plot was more or less solid dark blue, with no visible motifs at all. (This was the case for all three texts, so we did not include any of the ten-dimension plots in the results pictured in figure 1.) By 20 dimensions, as pictured in figure 1, the criss-cross pattern had appeared in much the same shape, but lacking a significant amount of detail. It is not until we get to 50 dimensions that it starts to more or less accurately resemble the patterns shown at 100 and 300 dimensions. Of the pale stripes that were obvious in the plot at 300 dimensions, only sentences 14, 16, and 53 become dark enough at 50 dimensions to appear cohesive—the overall pattern remains intact. The 70-dimension plot was, almost identical to the 50-dimension plot, and this was the case for all three texts, so we did not include any of the 70-dimension plots in the results pictured in figure 1. At 100 dimensions, the patterns are slightly more defined than at 50, but overall, it is clear that an analyst would reach the same conclusions about the text, whether they were guided by the plot at 50, 100, or 300 dimensions.

5.3 Education report

The most prominent motifs in the 300-dimension education report recurrence plot are the three examples of local (intra-paragraph) cohesion, which present as darker triangles along the outside edge of the plot:

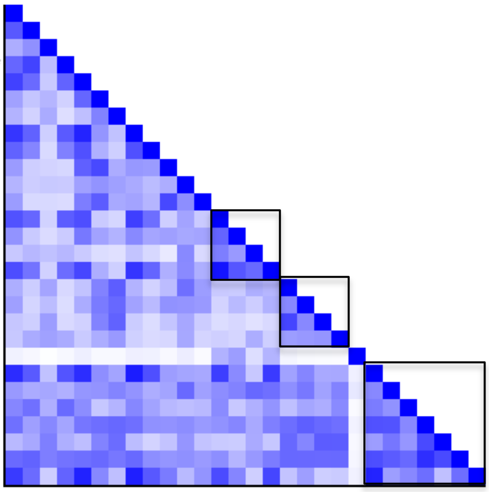


Figure 3. Local cohesion motifs in 300-dimension Lex plot of the Education report

By its original definition, a paragraph is the sustained development of a single idea (Rodgers, 1965), so it stands to reason that the sentences within a paragraph should share more concepts with each other than with sentences in other paragraphs. In this instance, however, the first two motifs represent just one paragraph, as well as the first sentence of the following paragraph. Examining this excerpt of the text offers some insight into why the plot may have divided this paragraph into two distinct motifs:

13. Deep Knowledge Highly effective teachers have a deep understanding of the subjects they teach.
14. These teachers have studied the content they teach in considerably greater depth than the level at which they currently teach and they have high levels of confidence in the subjects they teach.
15. Their deep content knowledge allows them to focus on teaching underlying methods, concepts, principles and big ideas in a subject, rather than on factual and procedural knowledge alone.
16. Highly effective teachers not only have deep knowledge of the subjects they teach, they also have deep understandings of how students learn those subjects (that is, pedagogical content knowledge).
17. They understand how learning typically progresses in a subject: for example, the skills and understandings that are pre-requisites for progress, and common paths of student learning.
18. They are familiar with the kinds of learning difficulties that some students experience and with appropriate interventions and available professional support for those difficulties.
19. And they are aware of common student misunderstandings and errors and know how to diagnose and address obstacles to further learning.
20. Targeted Teaching The most important single factor influencing learning is what the learner already knows.

In sentences 13 to 16, the subject 'teachers' (or variations thereof) is repeated throughout. In sentences 17 to 19, however, 'teachers' is replaced by the pronoun 'they', and the focus shifts to 'students' or 'learners'. Sentence 20 continues the theme, using both 'learner'/'learning' and 'teaching'.

The third local cohesion motif is formed mostly by the last two paragraphs, which together form a sub-section of the report entitled 'Targeted teaching'—though the section begins two sentences before the motif. When the low cohesion stripe discussed below (see figure 4) is accounted for, however, this motif aligns very well with the deliberate sectioning of the text.

The other noticeable motif in the Education report is the pale stripe in sentence 21 (figure 4), which, as in the Domestic violence report, is seemingly evidence of a short sentence containing few content words ("Ascertain this and teach him accordingly"), rather than a true example of low cohesion.

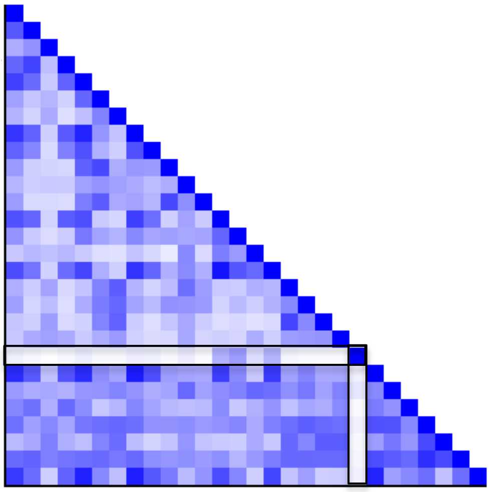


Figure 4. Low cohesion stripe motif in 300-dimension Lex plot of the Education report

Figure 1 demonstrates that the motifs for this text begin to disappear at 20 dimensions—whereas at 50 dimensions, the motifs in the 100- and 300-dimension plots are darker, but still clearly visible. Again, the threshold appears to be 50 dimensions.

5.4 Midwives report

The midwives report plot, at 300 dimensions, shows a dense introductory stripe, which is formed by the first two paragraphs of the text:

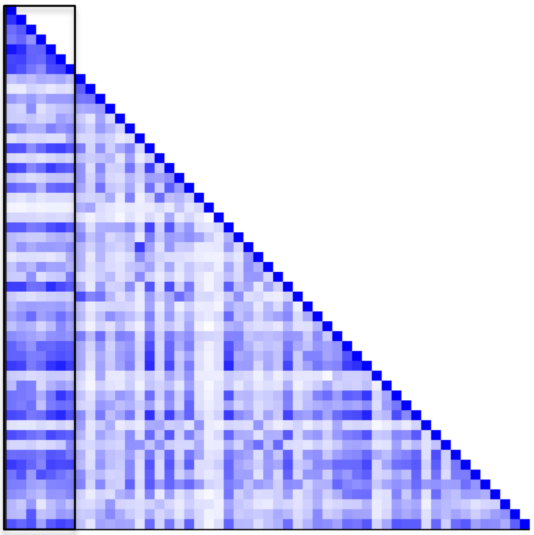


Figure 5. Introductory stripe motif in the 300-dimension Lex plot of the Midwives report

Although in the original document this text was split into two segments, together they can broadly be seen to represent the introductory section of the text, in that they set out the document's purpose and introduce and define the key terms heavily used throughout the rest of the document ('continuity', 'midwifery', 'care', 'birth', 'women', 'models', 'work', and variations thereof). The real business of the text is conducted after these two sections. Therefore, it is not surprising to see that it shows a greater level of cohesion both locally—within itself—and globally, with the entire rest of the document.

The second motif of interest is the two distinct pale stripes at sentences 21-23 and 39-40, signalling a group of sentences that do not share many concepts with either those preceding or following them. This pattern flags the possibility of low cohesion.

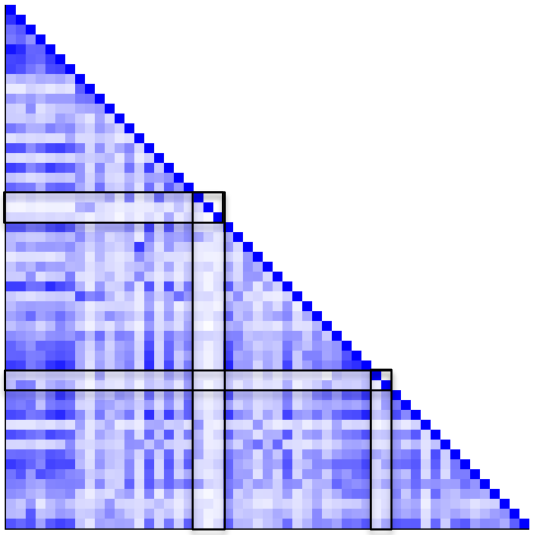


Figure 6. Low cohesion stripe motifs in 300-dimension plot of Midwives report

The full text of these two sentences are as follows:

Stripe 1

21. Communication within and beyond the service builds collaboration and understanding.

22. Engagement of stakeholders helps align expectations and manage divergent motivations.

23. 5. A guide to implementation

Stripe 2

39. This requires a different philosophy and skill set.

40. Relationships with women are close, continuous (sometimes for more than one baby), responsive to women's needs and very effective in supporting women's ability to birth and mother.

The palest stripes are again red herrings, caused by short sentences with few content words (sentences 23 and 39). The remaining sentences, especially 21 and 22, use a high proportion of abstract terms such as 'communication, 'collaboration', 'understanding', 'expectations', 'motivations', rather than the specific terms that more routinely occur throughout the text (variations of 'midwives', 'continuity models', 'birth', and 'women').

As with the plots for the other two texts, the motifs that are readily apparent at 300 dimensions hold steady until 20 dimensions, at which point they disappear completely. At 50 dimensions, it is likely that an analyst would reach the same conclusions as they would at 100 or 300 dimensions, but this would not be possible at 20.

6 Discussion and Conclusion

It is evident that, across all three texts, the visual motifs created by Lex start to become apparent and consistent at 50 dimensions. They are arguably a little clearer at 100 dimensions, and may even begin to fade out again at 300 dimensions. This finding is also supported in the quantitative data in Table 1, which shows that, for all three text-sample-and-corpora pairs, the absolute difference between 20 and 50 dimensions is much greater than between 50 and 100, or 100 and 300 dimensions.

This finding has implications for the original stated problem of whether allowing users to upload a custom corpus to a visual language analysis tool is a viable option. Using a MacBook Air running OSX Yosemite version 10.10.3 with a 1.7 GHz Intel Core i7 processor and 8GB of memory, the average processing time to build the semantic space from the corpus with 50 dimensions retained, analyse the input text, and render the visualisation for each of our three samples was 10.48 seconds, which we consider a reasonable time for commercial deployment. This suggests that allowing users to upload a custom corpus *is*, in fact, viable. Increasing the number of dimensions retained to 100 possibly brings with it a very small gain in performance, but a significant increase in processing time, given that the LSA algorithm utilises Singular Value Decomposition, which has an order O(n^3) complexity, where n is the number of dimensions.

Obviously, the findings outlined here are limited by a reliance on our own perception of the presence or absence of visual motifs. The next step will be to repeat this exercise on multiple texts, under controlled conditions involving external participants. We also plan to conduct further research exploring the effect of the size and specificity of the corpus.

7 References

Austen, L., Gilbert, B., & Mitchell, R. (2001). Plain English—an Ethical Issue? *Legal Ethics*, *4*(1), 5–7.

Biber, D. (1993). Representativeness in corpus design. *Literary and Linguistic Computing*, *8*(4), 243–257.

Bradford, R. B. (2008). An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 153–162). ACM.

Efron, M. (2005). Eigenvalue-based model selection during latent semantic indexing. *Journal of the American Society for Information Science and Technology*, *56*(9), 969–988.

Evangelopoulos, N., Zhang, X., & Prybutok, V. R. (2012). Latent semantic analysis: five methodological recommendations. *European Journal of Information Systems*, *21*(1), 70–86.

Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, *25*(2-3), 285–307. http://doi.org/10.1080/01638539809545029

Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, *36*(2), 193–202.

Kintsch, W., Patel, V. L., & Ericsson, K. A. (1999). The role of long-term working memory in text comprehension. *Psychologia*, *42*(4), 186–198.

Koester, A. (2010). Building small specialised corpora. In A. O'Keeffe & M. McCarthy (Eds.), *The Routledge Handbook of Corpus Linguistics* (pp. 66–79). Abingdon, UK: Taylor & Francis.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2-3), 259–284.

Lerman, K. (1999). Document clustering in reduced dimension vector space. *Unpublished Manuscript*. Retrieved from http://patwa.googlecode.com/svn-history/r15/trunk/docs/ref/Lerman99.pdf

Masters, C. (2009). *A shared challenge: Improving literacy, numeracy, and science skills in Queensland Primary Schools*. Australian Council for Educational Research. Retrieved from http://education.qld.gov.au/mastersreview/pdfs/final-report-masters.pdf

McCarthy, P. M., Briner, S. W., Rus, V., & McNamara, D. S. (2007). Textual signatures: Identifying text-types using latent semantic analysis to measure the cohesion of text structures. In *Natural language processing and text mining* (pp. 107–122). Springer.

McNamara, D. S., Kintsch, E., Songer, N. B., & Kintsch, W. (1996). Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction*, *14*(1), 1–43.

Queensland Government. (2012). *Delivering continuity of midwifery care to Queensland Women: A guide to implementation*. Brisbane, Queensland. Retrieved from http://www.qcmb.org.au/media/pdf/Midwives%20Imp%20guide_web.pdf

Rodgers, P. C. (1965). Alexander Bain and the rise of the organic paragraph. *Quarterly Journal of Speech*, *51*(4), 399–408. http://doi.org/10.1080/00335636509382739

Special Taskforce on Domestic and Family Violence in Queensland. (2015). *Not Now, Not Ever: Putting an end to domestic violence in Queensland*. Retrieved from https://www.qld.gov.au/community/documents/getting-support-health-social-issue/dfv-report-vol-one.pdf

Wade-Stein, D., & Kintsch, E. (2004). Summary Street: Interactive Computer Support for Writing. *Cognition and Instruction*, *22*(3), 333–362. http://doi.org/10.1207/s1532690xci2203_3

Ware, C. (2013). Foundations for an Applied Science of Data Visualization. In *Information Visualization* (pp. 1–30). Elsevier. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/B9780123814647000016

Watson, R. P., & Lynch, T. D. (1998). Plain English and Public Administration. *Public Policy and Administration*, *13*(1), 107–114. http://doi.org/10.1177/095207679801300108

Domain Adaption of Named Entity Recognition to Support Credit Risk Assessment

Julio Cesar Salinas Alvarado **Karin Verspoor** **Timothy Baldwin**
Department of Computing and Information Systems
The University of Melbourne
Australia
jsalinas@student.unimelb.edu.au
karin.verspoor@unimelb.edu.au
tb@ldwin.net

Abstract

Risk assessment is a crucial activity for financial institutions because it helps them to determine the amount of capital they should hold to assure their stability. Flawed risk assessment models could return erroneous results that trigger a misuse of capital by banks and in the worst case, their collapse. Robust models need large amounts of data to return accurate predictions, the source of which is text-based financial documents. Currently, bank staff extract the relevant data by hand, but the task is expensive and time-consuming. This paper explores a machine learning approach for information extraction of credit risk attributes from financial documents, modelling the task as a named-entity recognition problem. Generally, statistical approaches require labelled data for learn the models, however the annotation task is expensive and tedious. We propose a solution for domain adaption for NER based on out-of-domain data, coupled with a small amount of in-domain data. We also developed a financial NER dataset from publicly-available financial documents.

1 Introduction

In the years 2007–2008, the GFC (Global Financial Crisis) affected a vast number of countries around the world, causing losses of around USD$33 trillion and the collapse of big-name banks (Clarke, 2010). Experts identified that one of the main causes of the GFC was the use of poor financial models in risk assessment (Clarke, 2010; news.com.au, 2010; Debelle, 2009).

Risk assessment helps banks to estimate the amount of capital they should keep at hand to promote their stability and at the same time to protect their clients. Poor risk assessment models tend to overestimate the capital required, leading banks to make inefficient use of their capital, or underestimate the capital required, which could lead to banks collapsing in a financial crisis.

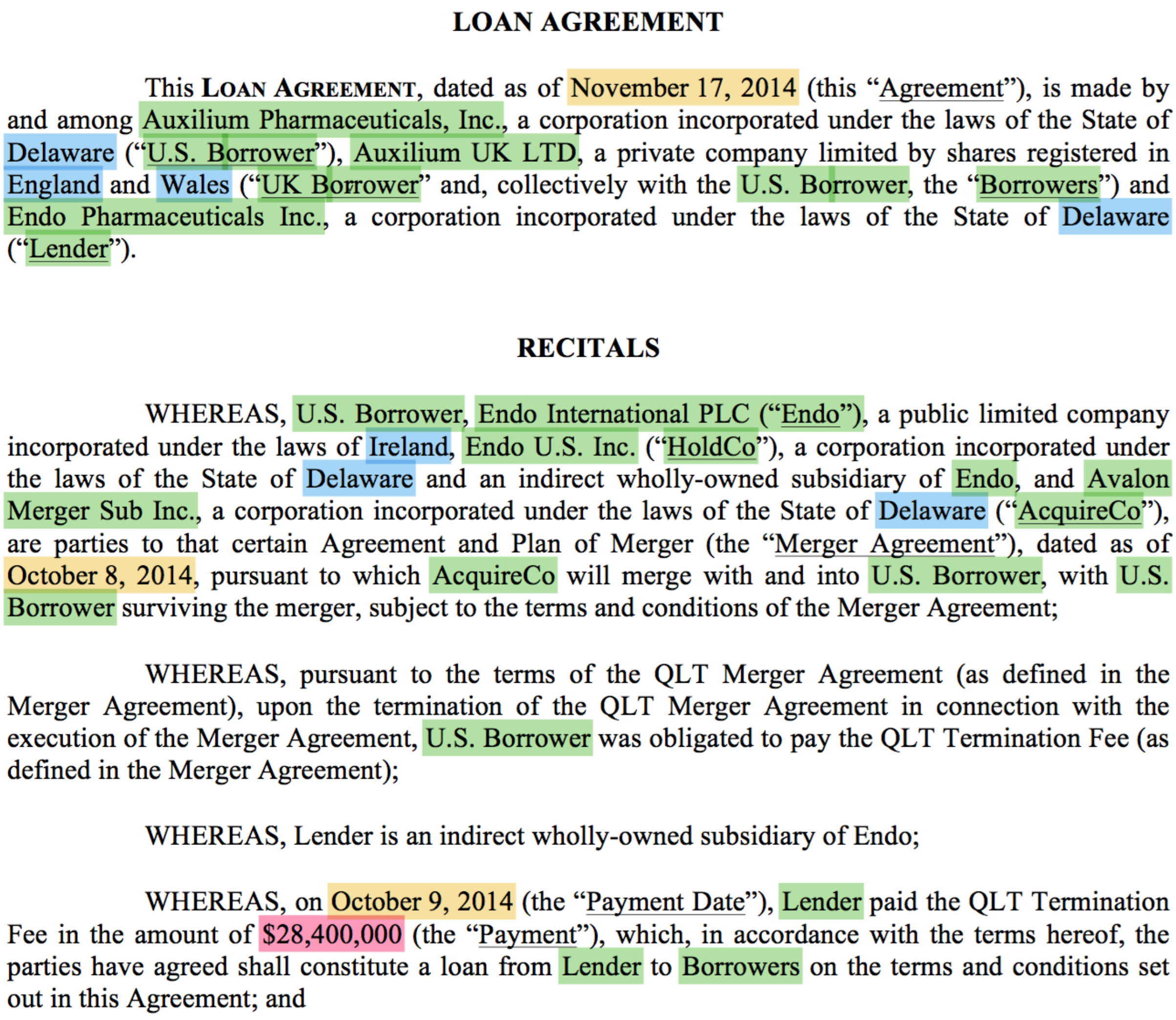
Financial documents such as contracts and loan agreements provide the information required to perform the risk assessment. These texts hold relevant details that feed into the assessment process, including: the purpose of the agreement, amount of loan, and value of collateral. Figure 1 provides a publicly available example of a loan agreement, as would be used in risk assessment.

Currently, bank staff manually extract the information from such financial documents, but the task is expensive and time-consuming for three main reasons: (1) all documents are in unstructured, textual form; (2) the volume of "live" documents is large, numbering in the millions of documents for a large bank; and (3) banks are continuously adding new information to the risk models, meaning that they potentially need to extract new fields from old documents they have previously analyzed.

Natural language processing (NLP) potentially offers the means to semi-automatically extract information required for risk assessment, in the form of named entity recognition (NER) over fields of interest in the financial documents. However, while we want to use supervised NER models, we also want to obviate the need for large-scale annotation of financial documents. The primary focus of this paper is how to build supervised NER models to extract information from financial agreements based on pre-existing out-of-domain data with partially-matching labelled data, and small amounts of in-domain data.

There are few public datasets in the financial domain, due to the privacy and commercial value of the data. In the interest of furthering research on information extraction in the financial domain, we

Julio Cesar Salinas Alvarado, Karin Verspoor and Timothy Baldwin. 2015. Domain Adaption of Named Entity Recognition to Support Credit Risk Assessment . In *Proceedings of Australasian Language Technology Association Workshop*, pages 84−90.



LOAN AGREEMENT

This **Loan Agreement**, dated as of November 17, 2014 (this "Agreement"), is made by and among Auxilium Pharmaceuticals, Inc., a corporation incorporated under the laws of the State of Delaware ("U.S. Borrower"), Auxilium UK LTD, a private company limited by shares registered in England and Wales ("UK Borrower" and, collectively with the U.S. Borrower, the "Borrowers") and Endo Pharmaceuticals Inc., a corporation incorporated under the laws of the State of Delaware ("Lender").

RECITALS

WHEREAS, U.S. Borrower, Endo International PLC ("Endo"), a public limited company incorporated under the laws of Ireland, Endo U.S. Inc. ("HoldCo"), a corporation incorporated under the laws of the State of Delaware and an indirect wholly-owned subsidiary of Endo, and Avalon Merger Sub Inc., a corporation incorporated under the laws of the State of Delaware ("AcquireCo"), are parties to that certain Agreement and Plan of Merger (the "Merger Agreement"), dated as of October 8, 2014, pursuant to which AcquireCo will merge with and into U.S. Borrower, with U.S. Borrower surviving the merger, subject to the terms and conditions of the Merger Agreement;

WHEREAS, pursuant to the terms of the QLT Merger Agreement (as defined in the Merger Agreement), upon the termination of the QLT Merger Agreement in connection with the execution of the Merger Agreement, U.S. Borrower was obligated to pay the QLT Termination Fee (as defined in the Merger Agreement);

WHEREAS, Lender is an indirect wholly-owned subsidiary of Endo;

WHEREAS, on October 9, 2014 (the "Payment Date"), Lender paid the QLT Termination Fee in the amount of $28,400,000 (the "Payment"), which, in accordance with the terms hereof, the parties have agreed shall constitute a loan from Lender to Borrowers on the terms and conditions set out in this Agreement; and

Figure 1: Example of a loan agreement. Relevant information that is used by risk assessment models is highlighted. The example is taken from a loan agreement that has been disclosed as part of an SEC hearing, available at `http://www.sec.gov/Archives/edgar/data/1593034/000119312514414745/d817818dex101.htm`

construct an annotated dataset of public-domain financial agreements, and use this as the basis of our experiments.

This paper describes an approach for domain adaption that includes a small amount of target domain data into the source domain data. The results obtained encourage the use of this approach in cases where the amount of target data is minimal.

2 Related Work

Most prior approaches to information extraction in the financial domain make use of rule-based methods. Farmakiotou et al. (2000) extract entities from financial news using grammar rules and gazetteers. This rule-based approach obtained 95% accuracy overall, at a precision and recall of 78.75%. Neither the number of documents in the corpus nor the number of annotated samples used in the work is mentioned, but the number of words in the corpus is 30,000 words for training and 140,000 for testing. The approach involved the creation of rules by hand; this is a time-consuming task, and the overall recall is low compared to other extraction methods.

Another rule-based approach was proposed by Sheikh and Conlon (2012) for extracting information from financial data (combined quarterly reports from companies and financial news) with the aim of assisting in investment decision-making.

The rules were based on features including exact word match, part-of-speech tags, orthographic features, and domain-specific features. After creating a set of rules from annotated examples, they tried to generalize the rules using a greedy search algorithm and also the Tabu Search algorithm. They obtained the best performance of 91.1% precision and 83.6% recall using the Tabu Search algorithm.

The approach of Farmakiotou et al. (2000) is similar to our approach in that they tried to address an NER problem with financial data. However, their data came from financial news rather than the financial agreements, as targeted in our work. The focus of Sheikh and Conlon (2012) is closer to that in this paper, in that they make use of both financial news and corporate quarterly reports. However, their extraction task does not consider financial contracts, which is the key characteristic of our problem setting.

Somewhat further afield — but related in the sense that financial agreements stipulate the legal terms of a financial arrangement — is work on information extraction in the legal domain. Moens et al. (1999) used information extraction to obtain relevant details from Belgian criminal records with the aim of generating abstracts from them. The approach takes advantage of discourse analysis to find the structure of the text and linguistic forms, and then creates text grammars. Finally, the approach uses a parser to process the document content. Although the authors do not present results, they argue that when applied to a test set of 1,000 criminal cases, they were able to identify the required information.

In order to reduce the need for annotation, we explore domain adaptation of an information extraction system using out-of-domain data and a small amount of in-domain data. Domain adaptation for named entity recognition techniques has been explored widely in recent years. For instance, Jiang and Zhai (2006) approached the problem by generalizing features across the source and target domain to way avoid overfitting. Mohit and Hwa (2005) proposed a semi-supervised method combining a naive Bayes classifier with the EM algorithm, applied to features extracted from a parser, and showed that the method is robust over novel data. Blitzer et al. (2006) induced a correspondence between features from a source and target domain based on structural correspondence learning over unlabelled target domain data. Qu et al. (2015) showed that a graph transformer NER model trained over word embeddings is more robust cross-domain than a model based on simple lexical features.

Our approach is based on large amounts of labelled data from a source domain and small amounts of labelled data from the target domain (i.e. financial agreements), drawing inspiration from previous research that has shown that using a modest amount of labelled in-domain data to perform transfer learning can substantially improve classifier accuracy (Duong et al., 2014).

3 Background

Named entity recognition (NER) is the task of identifying and classifying token-level instances of named entities (NEs), in the form of proper names and acronyms of persons, places or organizations, as well as dates and numeric expressions in text (Cunningham, 2005; Abramowicz and Piskorski, 2003; Sarawagi, 2008). In the financial domain, example NE types are LENDER, BORROWER, AMOUNT, and DATE.

We build our supervised NER models using conditional random fields (CRFs), a popular approach to sequence classification (Lafferty et al., 2001; Blunsom, 2007). CRFs model the conditional probability $p(s|o)$ of labels (states) s given the observations o as in Equation 1, where t is the index of words in observation sequence o, each k is a feature, w_k is the weight associated with the feature k, and $Z_w(o)$ is a normalization constant.

$$p(s|o) = \frac{\exp(\sum_t \sum_k w_k f_k(s_{t-1}, s_t, o, t))}{Z_w(o)} \quad (1)$$

4 Methods

4.1 Data

In order to evaluate NER over financial agreements, we annotated a dataset of financial agreements made public through U.S. Security and Exchange Commission (SEC) filings. Eight documents (totalling 54,256 words) were randomly selected for manual annotation, based on the four NE types provided in the CoNLL-2003 dataset: LOCATION (LOC), ORGANISATION (ORG), PERSON (PER), and MISCELLANEOUS (MISC). The annotation was carried out using the Brat annotation tool (Stenetorp et al., 2012). All documents were pre-tokenised and part-of-speech (POS) tagged using NLTK (Bird et al., 2009). As part of the annotation, we automatically tagged all instances of the tokens *lender* and *borrower* as being of entity type PER. We have made this dataset available in CoNLL format for research purposes at: `http://people.eng.unimelb.edu.au/tbaldwin/resources/finance-sec/`.

For the training set, we use the CoNLL-2003 English data, which is based on Reuters newswire data and includes part-of-speech and chunk tags (Tjong Kim Sang and De Meulder, 2003).

The eight financial agreements were partitioned into two subsets of five and three documents, which we name "FIN5" and "FIN3", respectively. The former is used as training data, while the latter is used exclusively for testing.

Table 1 summarizes the corpora.

4.2 Features

For all experiments, we used the CRF++ toolkit (Kudo, 2013), with the following feature set (optimized over the CoNLL-2003 development set):

- Word features: the word itself; whether the word starts with an upper case letter; whether the word has any upper case letters other than the first letter; whether the word contains digits or punctuation symbols; whether the word has hyphens; whether the word is all lower or upper case.
- Word shape features: a transformation of the word, changing upper case letters to *X*, lower case letters to *x*, digits to *0* and symbols to *#*.
- Penn part-of-speech (POS) tag.
- Stem and lemma.
- Suffixes and Prefixes of length 1 and 2.

4.3 Experimental Setup and Results

We first trained and tested directly on the CoNLL-2003 data, resulting in a model with a precision of 0.833, recall of 0.824 and F1-score of 0.829 (**Experiment1**), competitive with the start-of-the-art for the task.

The next step was to experiment with the financial data. For that, first we applied the CoNLL-2003 model directly to FIN3. Then, in order to improve the results for the domain adaption, we trained a new model using the CONLL +FIN5 data set, and test this model against the FIN3 dataset.

A summary of the experimental results over the financial data sets is presented in Table 2.

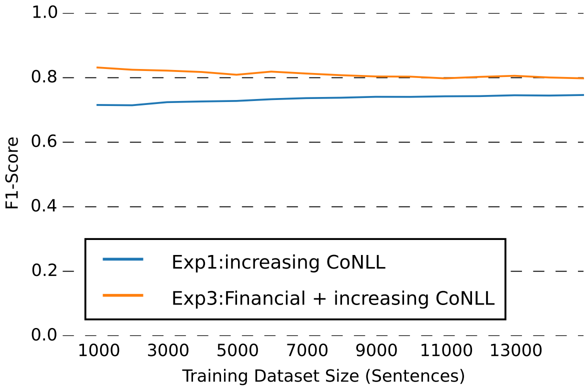


Figure 2: Learning curves showing the F-Score as more CONLL data is added for Experiment1 and Experiment3. Experiment3 starts in FIN5 and incrementally adding CONLL data.

5 Discussion

Table 2 summarizes the results of directly applying the model obtained by training only over out-of-domain data to the two financial data sets. The difference in the domain composition of the CONLL data (news) and the financial documents can be observed in these results. With out-of-domain test data, a precision of 0.247 and a recall of 0.132 (**Experiment2**) was observed, while testing with in-domain data achieved a precision of 0.833 and recall of 0.824 (**Experiment1**).

As a solution to the difference in the nature of the sources in the context of limited annotated in-domain data, we experimented with simple domain adaptation, by including into the source domain (CONLL) data a small amount of the target domain data — i.e. including data from FIN5— generating a new training data set (CONLL +FIN5). When trained over this combined data set, the results increased substantially, obtaining a precision of 0.828, recall of 0.770 and F-score of 0.798 (**Experiment3**).

As additional analysis, in Figure 2, we plot learning curves based on F-score obtained for Experiment2 and Experiment3 as we increase the training set (in terms of the number of sentences). We can see that the F-score increases slightly with increasing amounts of pure CONLL data (Experiment2), but that in the case of the mixed training data (Experiment3), the results actually drop as we add more CONLL data.

Figure 3 shows the learning curves for Experiment3 and Experiment4, as we add more financial

Name	Description
CoNLL	CoNLL-2003 training data
CoNLL$_{test}$	CoNLL-2003 test data
CoNLL +Fin5	CoNLL-2003 training data + five financial agreements
Fin5	Five financial agreements
Fin3	Three financial agreements

Table 1: Description of the data sets used.

Name	Training Data	Test Data	P	R	F1
Experiment1	CoNLL	CoNLL$_{test}$	0.833	0.824	0.829
Experiment2	CoNLL	Fin3	0.247	0.132	0.172
Experiment3	CoNLL +Fin5	Fin3	0.828	0.770	0.798
Experiment4	Fin5	Fin3	0.944	0.736	0.827

Table 2: Results of testing over the financial data sets.

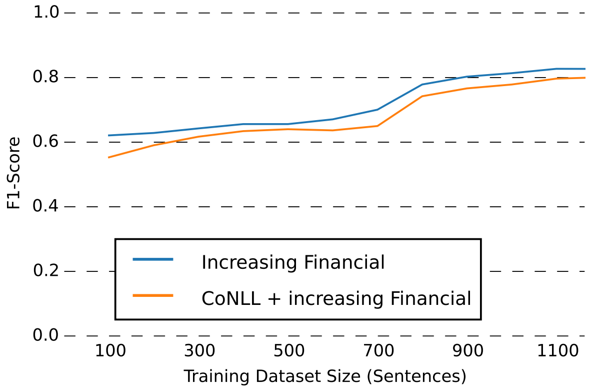


Figure 3: Learning curves showing the F-score as more financial training data is added for Experiment3 and Experiment 4.

data. Here, in the case of Experiment3, we start out with all of the CoNLL data, and incrementally add Fin5. We can see that the more financial data we add, the more the F-score improves, with a remarkably constant absolute difference in F-score between the two experiments for the same amount of in-domain data. That is, even for as little as 100 training sentences, the CoNLL data degrades the overall F-score.

Confusion matrices for the results of the predictions of **Experiment3** are shown in Table 3.

Analysis of the errors in the confusion matrix reveals that the entity type MISC has perfect recall over the financial dataset. Following MISC, PER is the entity type with the next best recall, at over 0.9. However, generally the model tends to suffer from a high rate of false positives for the entities LOC and ORG, affecting the precision of those classes and the overall performance of the model.

One interesting example of error in the output of the model is when the tokens refer to an address. One example is the case of *40 Williams Street*, where the correct label is LOC but the model predicts the first token (*40*) to be NANE and the other two tokens to be an instance of PER (i.e. *Williams Street* is predicted to be a person).

In the model generated with just the CoNLL data, one notable pattern is consistent false positives on tokens with initial capital letters; for example, the model predicts both *Credit Extensions* and *Repayment Period* to be instances of ORG, though in the gold standard they don't belong to any entity type. This error was reduced drastically through the addition of the in-domain financial data in training, improving the overall performance of the model.

Ultimately, the purely in-domain training stratagem in Experiment4 outperforms the mixed data setup (Experiment3), indicating that domain context is critical for the task. Having said that, the results of our study inform the broader question of out-of-domain applicability of NER models. Furthermore, they point to the value of even a small amount of in-domain training data (Duong et al., 2014).

6 Conclusions

Risk assessment is a crucial task for financial institutions such as banks because it helps to estimate the amount of capital they should hold to promote their stability and protect their clients. Manual extraction of relevant information from text-

		Predicted					
		LOC	MISC	ORG	PER	O	Recall
	LOC	**20**	0	3	2	14	0.513
	MISC	0	**7**	0	0	0	1.000
Actual	ORG	0	0	**16**	0	40	0.286
	PER	0	0	0	**202**	14	0.935
	NANE	12	2	24	8	–	
	Precision	0.625	0.778	0.372	0.953		

Table 3: Confusion matrix for the predictions over FIN3 using the model from Experiment3, including the precision and recall for each class ("NANE" = Not a Named Entity).

based financial documents is expensive and time-consuming.

We explored a machine learning approach that modelled the extraction task as a named entity recognition task. We used a publicly available non-financial dataset as well as a small number of annotated publicly available financial documents. We used a conditional random field (CRF) to label entities. The training process was based on data from CoNLL-2003 which had annotations for the entity types PER (person), MISC (miscellaneous), ORG (organization) and LOC (location). We then assembled a collection of publicly-available loan agreements, and manually annotated them, to serve as training and test data. Our experimental results showed that, for this task and our proposed approach, small amounts of in-domain training data are superior to large amounts of out-of-domain training data, and furthermore that supplementing the in-domain training data with out-of-domain data is actually detrimental to overall performance.

In future work, we intend to test this approach using different datasets with an expanded set of entity types specific to credit risk assessment, such as values and dates. An additional step would be carry out extrinsic evaluation of the output of the model in an actual credit risk assessment scenario. As part of this, we could attempt to identify additional features for risk assessment, beyond what is required by the financial authorities.

References

Witold Abramowicz and Jakub Piskorski. 2003. Information extraction from free-text business documents. In Stephanie Becker, editor, *Effective Databases for Text & Document Management*, pages 12–23. IRM Press.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python — Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Sebastopol, USA.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia.

Philip Blunsom. 2007. *Structured classification for multilingual natural language processing*. Ph.D. thesis, University of Melbourne Melbourne, Australia.

Thomas Clarke. 2010. Recurring crises in Anglo-American corporate governance. *Contributions to Political Economy*, 29(1):9–32.

Hamish Cunningham. 2005. Information extraction, automatic. In *Encyclopedia of Language and Linguistics*, pages 665–677. Elsevier, 2nd edition.

Guy Debelle. 2009. Some effects of the global financial crisis on australian financial markets. http://www.rba.gov.au/speeches/2009/sp-ag-310309.html.

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. 2014. What can we get from 1000 tokens? a case study of multilingual POS tagging for resource-poor languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 886–897, Doha, Qatar.

Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. 2000. Rule-based named entity recognition for Greek financial texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*, pages 75–78, Patras, Greece.

Jing Jiang and ChengXiang Zhai. 2006. Exploiting domain structure for named entity recognition. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 74–81, New York, USA.

Taku Kudo. 2013. CRF++: Yet another CRF toolkit. `https://taku910.github.io/crfpp/`. Accessed 26 May, 2015.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, USA.

Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. 1999. Information extraction from legal texts: the potential of discourse analysis. *International Journal of Human-Computer Studies*, 51(6):1155–1171.

Behrang Mohit and Rebecca Hwa. 2005. Syntax-based semi-supervised named entity tagging. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, pages 57–60, Ann Arbor, USA.

news.com.au. 2010. Poor risk assessment 'led to global financial crisis'. `http://goo.gl/f92sv8`. Accessed 10 Nov, 2015.

Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, Nathan Schneider, and Timothy Baldwin. 2015. Big data small data, in domain out-of domain, known word unknown word: The impact of word representations on sequence labelling tasks. In *Proceedings of the 19th Conference on Natural Language Learning (CoNLL-2015)*, pages 83–93, Beijing, China.

Sunita Sarawagi. 2008. Information Extraction. *Foundations and Trends in Databases*, 1(3):261–377.

Mahmudul Sheikh and Sumali Conlon. 2012. A rule-based system to extract financial information. *Journal of Computer Information Systems*, 52(4):10–19.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: A web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Do POS Tags Help to Learn Better Morphological Segmentations?

Kairit Sirts
Department of Computing
Macquarie University
Australia
kairit.sirts@gmail.com

Mark Johnson
Department of Computing
Macquarie University
Australia
mark.johnson@mq.edu.au

Abstract

The utility of using morphological features in part-of-speech (POS) tagging is well established in the literature. However, the usefulness of exploiting information about POS tags for morphological segmentation is less clear. In this paper we study the POS-dependent morphological segmentation in the Adaptor Grammars framework. We experiment with three different scenarios: without POS tags, with gold-standard tags and with automatically induced tags, and show that the segmentation F1-score improves when the tags are used. We show that the gold-standard tags lead to the biggest improvement as expected. However, using automatically induced tags also brings some improvement over the tag-independent baseline.

1 Introduction

Linguistially, part-of-speech (POS) tagging and morphology are closely related and this relation has been heavily exploited in both supervised and unsupervised POS tagging. For instance, the supervised Stanford tagger (Toutanova et al., 2003) as well as some unsupervised POS taggers (Berg-Kirkpatrick et al., 2010; Lee et al., 2010) use character prefix and/or suffix features, while the model by Christodoulopoulos et al. (2011) makes use of suffixes learned with an unsupervised morphological segmentation model.

There have been some attempts to exploit the relation in the opposite direction to learn the segmentations dependent on POS tags. For instance, the segmentation procedures described by Freitag (2005) and Can and Manandhar (2009) find the syntactic clusters of words and then perform morphology learning using those clusters. Few works have included a small number of syntactic classes directly into the segmentation model (Goldwater et al., 2006; Lee et al., 2011). However, Goldwater et al. (2006) only trains the model on verbs, which means that the classes model different verb paradigms rather than POS tags. Secondly, the model is never evaluated in a single class configuration and thus it is not known whether incorporating those classes gives any actual improvement. The results of Lee et al. (2011) show small improvements when the POS-word component (a bigram HMM) is incorporated into the model. However, the number of syntactic categories they learn is only 5, which is smaller than the number of main POS categories in most annotated corpora. Moreover, the main gain in the segmentation F-score is obtained by modeling the agreements between adjacent words, rather than exploiting the relation to syntactic classes.

Another line of previous work has attempted to model the POS tags and morphological segmentations jointly in an unsupervised model (Can, 2011; Sirts and Alumäe, 2012; Frank et al., 2013). However, the results presented in those papers fail to demonstrate clearly the utility of using the tag information in segmentation learning over the scenario where the tags are missing.

The goal of this paper is to explore the relation between POS tags and morphological segmentations and in particular, to study if and how much the POS tags help to learn better segmentations. We start with experiments learning segmentations without POS tags as has been standard in previous literature (Goldsmith, 2001; Creutz and Lagus, 2007; Sirts and Goldwater, 2013) and then add the POS information. We first add the information about gold-standard tags, which provides a kind of upper bound of how much the segmentation accuracy can gain from POS information. Secondly, we also experiment with automatically induced tags. We expect to see that gold-standard POS tags improve the segmentation accuracy and that induced tags

Kairit Sirts and Mark Johnson. 2015. Do POS Tags Help to Learn Better Morphological Segmentations? . In *Proceedings of Australasian Language Technology Association Workshop*, pages 91−100.

are also helpful. The results of these experiments can be informative to whether directing effort into developing joint unsupervised models for POS tagging and segmentation is justified, or whether the efforts of exploiting synergies in morphology learning should be focused elsewhere.

We define the segmentation model in the Adaptor Grammars framework (Johnson et al., 2007) that has been previously successfully applied to learning morphological segmentations (Johnson, 2008; Sirts and Goldwater, 2013). In fact, we will use some of the grammars defined by Sirts and Goldwater (2013) but enrich the grammar rules with information about POS tags. Our POS-dependent grammars are inspired by the grammars used to learn topic models (Johnson, 2010), which have separate rules for each topic. In a similar fashion we will have a separate set of rules for each POS tag.

We conduct experiments both in English and Estonian—a morphologically rich inflective and agglutinative language—and show that the grammars exploiting information about the gold-standard POS tags indeed learn better morphological segmentations in terms of F1-score. The gain in scores when compared to the tag-independent segmentations is up to 14%, depending on the language and the grammar. When the model uses automatically induced tags, the learned segmentations in English are still better than the tag-independent baseline, but the differences in scores are smaller, reaching up to 11% absolute improvement. Although the scores show improvements in Estonian as well, the closer inspection of segmentations of different POS category words reveals that in most cases there are no major differences between segmentations learned with and without tags.

The rest of the paper is organized as follows. In section 2 we briefly introduce the Adaptor Grammars framework, section 3 describes the tag-dependent grammars used in experiments. Section 4 lists the experimental scenarios. In section 5 we describe the experimental setup. Section 6 presents the results, followed by the discussion in section 7, section 8 concludes the paper.

2 Adaptor Grammars

Adaptor Grammars (AG) (Johnson et al., 2007) is a non-parametric Bayesian framework for learning latent structures over sequences of strings. In the current context, the sequence of strings is a sequence of characters making up a word, and the latent structures of interest are the morphemes.

An AG consists of two components: a probabilistic context-free grammar (PCFG) that can generate all possible latent structures for the given inputs, and a Pitman-Yor process (PYP) adaptor function that transforms the probabilities of the parse trees in such a way that the probabilities of the frequently occurring subtrees are much higher than they would be under the PCFG model.

A simple morphological grammar for the AG model could be (Sirts and Goldwater, 2013):

$$\begin{aligned}\text{Word} &\rightarrow \text{Morph}^{+}\\ \underline{\text{Morph}} &\rightarrow \text{Char}^{+},\end{aligned}$$

where each word consists of one or more morphemes and each morpheme is a sequence of characters. The grammar here uses an abbreviated notation for denoting the recursive rules and thus the first rule is a short-hand writing for:

$$\begin{aligned}\text{Word} &\rightarrow \text{Morphs}\\ \text{Morphs} &\rightarrow \text{Morph}\\ \text{Morphs} &\rightarrow \text{Morph Morphs}\end{aligned}$$

The underline denotes the adapted non-terminals, i.e. the sub-trees rooted in those non-terminals are cached by the model and their probabilities are computed according to the PYP. In the given example the Morph non-terminal is adapted, which means that the model prefers to re-generate the same subtrees denoting the morphemes repeatedly.

We use in our experiments an existing AG implementation[1], the technical details of this implementation are described in (Johnson and Goldwater, 2009).

3 POS-dependent Grammars

The POS-dependent grammars are inspired by the grammars that have been used to learn topic models (Johnson, 2010). Whereas the topic modeling grammars have one rule for every latent topic, the POS-dependent grammars have one rule for each possible tag, which enables the model to cache the subtrees corresponding to morphemes in words with specific syntactic category.

[1] available from http://web.science.mq.edu.au/~mjohnson/Software.htm

Consider for instance a tagset that contains three tags: verb, noun and adjective. Then, in order to make the simple morpheme sequences generating grammar shown in the previous section to be POS-dependent, the rules for each POS tag have to be replicated:

$$\begin{aligned} \text{Word} &\rightarrow \text{Noun Morph}^{+}_{\text{Noun}} \\ \text{Word} &\rightarrow \text{Verb Morph}^{+}_{\text{Verb}} \\ \text{Word} &\rightarrow \text{Adj Morph}^{+}_{\text{Adj}} \\ \underline{\text{Morph}_{\text{Noun}}} &\rightarrow \text{Char}^{+} \\ \underline{\text{Morph}_{\text{Verb}}} &\rightarrow \text{Char}^{+} \\ \underline{\text{Morph}_{\text{Adj}}} &\rightarrow \text{Char}^{+}, \end{aligned}$$

Each rule rooted in Word now first generates a non-terminal that corresponds to a particular POS tag and a sequence of POS-specific morphemes. In order to make the grammar complete, we also add rules that generate the terminal symbols corresponding to specific POS tags. We add an underscore to the terminal symbols corresponding to tags to distinguish them from other terminal symbols that are used to generate the words themselves.

$$\begin{aligned} \text{Noun} &\rightarrow \text{N\_} \\ \text{Verb} &\rightarrow \text{V\_} \\ \text{Adj} &\rightarrow \text{A\_} \end{aligned}$$

We experiment with three different grammars that generate POS-dependent morphological segmentations. The first two of them, **MorphSeq** and **SubMorph** are essentially the same as the ones used for morphological segmentation in (Sirts and Goldwater, 2013). The third one, **CollocMorph**, adds another layer of latent structure on top of morphemes to model morpheme collocations. All three grammars are made tag-dependent by replicating the relevant rules by using tag-specific non-terminals as explained above.

The **MorphSeq**, which was also given as an example in Section 2, is the simplest grammar that just generates each word as a sequence of morphemes. It is essentially a unigram morphology model. The tag-dependent version we used is the following:

$$\begin{aligned} \text{Word} &\rightarrow \text{Tag Morph}^{+}_{\text{tag}} && \text{for } \forall \text{ tag} \in T \\ \underline{\text{Morph}_{\text{tag}}} &\rightarrow \text{Morph} && \text{for } \forall \text{ tag} \in T \\ \text{Tag} &\rightarrow \tau && \text{for } \forall \tau \in \mathcal{T} \\ \underline{\text{Morph}} &\rightarrow \text{Char}^{+} \end{aligned}$$

Here, T is the set of non-terminal symbols denoting different tags. For instance, this set could be $\{N, V, A\}$ denoting nouns, verbs and adjectives. $\mathcal{T}$ is the corresponding set of tag terminal symbols. Each tag-specific Morph non-terminal also generates a general back-off Morph non-terminal which is shared between all tags. This is desirable because words with different syntactic categories may share the same set of stems. Also, some suffixes are reused across different syntactic categories, either due to agreement or polysemy.

The **SubMorph** grammar adds an additional level of latent structure below the morphemes by generating each morpheme as a sequence of sub-morphemes. In (Sirts and Goldwater, 2013), this was shown to improve the segmentation results considerably. We define the morphemes as tag-specific and specify that sub-morphemes are shared across all tags. In preliminary experiments we also tried to make sub-morphemes tag-specific but this grammar did not produce good results.

$$\begin{aligned} \text{Word} &\rightarrow \text{Tag Morph}^{+}_{\text{tag}} && \text{for } \forall \text{ tag} \in T \\ \underline{\text{Morph}_{\text{tag}}} &\rightarrow \text{Morph} && \text{for } \forall \text{ tag} \in T \\ \text{Tag} &\rightarrow \tau && \text{for } \forall \tau \in \mathcal{T} \\ \underline{\text{Morph}} &\rightarrow \text{SubMorph}^{+} \\ \underline{\text{SubMorph}} &\rightarrow \text{Char}^{+} \end{aligned}$$

The third grammar, **CollocMorph**, extends the SubMorph grammar and adds another layer of morpheme collocations on top of Morphs. In this grammar both morpheme collocations and morphemes are tag-specific while sub-morphemes are again general:

$$\begin{aligned} \text{Word} &\rightarrow \text{Tag Colloc}^{+}_{\text{tag}} && \text{for } \forall \text{ tag} \in T \\ \underline{\text{Colloc}_{\text{tag}}} &\rightarrow \text{Morph}^{+}_{\text{tag}} && \text{for } \forall \text{ tag} \in T \\ \underline{\text{Morph}_{\text{tag}}} &\rightarrow \text{Morph} && \text{for } \forall \text{ tag} \in T \\ \text{Tag} &\rightarrow \tau && \text{for } \forall \tau \in \mathcal{T} \\ \underline{\text{Morph}} &\rightarrow \text{SubMorph}^{+} \\ \underline{\text{SubMorph}} &\rightarrow \text{Char}^{+} \end{aligned}$$

4 Experimental Scenarios

In order to assess how much the syntactic tags affect the accuracy of the morphological segmentatations, we conducted experiments using four different scenarios:

1. POS-independent morphological segmentation;
2. POS-dependent morphological segmentation using gold-standard tags;
3. POS-dependent segmentation using syntactic clustering learned with an unsupervised model;
4. POS-dependent segmentation using randomly generated tags.

The first scenario does not use any tags at all and is thus the standard setting used in previous work for conducting unsupervised morphological segmentation. This is the baseline we expect the other, tag-dependent scenarios to exceed.

The second scenario, which uses gold-standard POS tags, is an oracle setting that gives an upper bound of how much the tags can help to improve the segmentation accuracy when using a particular segmentation model. Hypothetically, there could exist tagging configurations, which improve the segmentations more than the oracle tags but in our experiments this was not the case.

The third scenario uses the tags learned with an unsupervised POS induction model. Our expectation here is that the segmentations learned with this scenario are better than the baseline without any tags and worse than using gold-standard tags. The experimental results presented later confirm that this is indeed the case.

The final scenario is the second baseline using tags generated uniformly at random. By evaluating this scenario we hope to show that not just *any* tagging configuration improves the segmentation results but the tags must really correspond at least to some extent to real syntactic tags.

5 Experimental Setup

We conduct experiments in both English and Estonian—a morphologically complex language belonging to Fenno-Ugric language group, using all four scenarios explained above and all three described grammars. AG is a stochastic model and thus it may produce slightly different results on different runs. Therefore, we run the AG in each setting consisting of the language-scenario-grammar triple for 10 times with different random initialisations. We run the sampler for 1000 iterations, after which we collect a single sample and aggregate the samples from all runs by using maximum marginal decoding (Johnson and Goldwater, 2009; Stallard et al., 2012). We use batch initialisation, table label resampling is turned on and all hyperparameters are inferred.

	English	**Estonian**
MTE types	8438	15132
Eval types	7659	15132
Eval nouns	3831	8162
Eval verbs	2691	4004
Eval adjectives	1629	3111

Table 1: The number of open class words (nouns, verbs and adjectives) used for training and evaluation.

5.1 Data

All experiments were conducted on English and Estonian parts of the Multext-East (MTE) corpus (Erjavec, 2004) that contains G. Orwell's novel ”1984”. The MTE corpora are morpho-syntactically annotated and the label of each word also contains the POS tag, which we can use in the oracle experiments that make use of gold-standard tags. However, the annotations do not include morphological segmentations. For Estonian, this text is also part of the morphologically disambiguated corpus,[2] which has been manually annotated and also contains segmentations. We use Celex (Baayen et al., 1995) as the source for English gold-standard segmentations, which have been extracted with the Hutmegs package (Creutz and Lindén, 2004). Although not all the words from the MTE English part are annotated in Celex, most of them do, which provides a reasonable basis for our evaluations.

We conduct experiments only on a subset of word types from the MTE corpora, in particular on nouns, verbs and adjectives only. These POS categories constitute open class words and thus are expected to contain the most morphological richness. The statistics about the number of word types in the training and evaluation sets as well as the number of words belonging to different POS categories for both English and Estonian are given in Table 1. The counts of nouns, verbs and adjectives

[2] http://www.cl.ut.ee/korpused/morfkorpus/index.php?lang=en

	English				Estonian			
	No POS	**Gold**	**Learned**	**Rand**	**No POS**	**Gold**	**Learned**	**Rand**
MorphSeq	51.4	54.3	55.7	52.5	48.1	53.2	52.5	49.1
SubMorph	63.3	69.6	68.1	64.3	66.5	66.5	64.3	65.5
CollocMorph	56.8	71.0	68.0	66.6	65.4	68.5	66.5	68.4

Table 2: F1-scores of all experiments in English and Estonian using different grammars and settings. **MorphSeq** generates sequences of morphemes, **SubMorph** adds the sub-morphemes, and **CollocMorph** adds the morpheme collocations. **No POS** are the models trained without tags, **Gold** uses goldstandard POS tags, **Learned** uses tags learned by an unsupervised POS induction model, and **Rand** uses randomly generated tags.

do not add up to the total number of evaluated word types because some of the words in the corpus are ambiguous and occur in different syntactic roles.

The automatically induced syntactic tags were learned with an unsupervised POS induction model (Sirts and Alumäe, 2012).[3] The main reason for choosing this model was the fact that it has been evaluated on the same MTE corpus we use for learning on both English and Estonian and has shown to produce reasonably good tagging results.

5.2 Input Format

For POS-independent segmentation we just train on the plain list of words. For tag-dependent experiments we have to reformat the input so that each word is preceded by its tag, which will be parsed by the left branch of the first rule in each grammar. For instance, the input for the tag-independent AG model for a noun `table` is just a sequence of characters separated by spaces:

```
t a b l e
```

However, for the tag-dependent model it has to be reformatted as:

```
N_ t a b l e,
```

where `N_` is the terminal symbol denoting the noun POS.

The tag assignments of the unsupervised POS induction model are just integer numbers and thus for instance, if the model has assigned a tag 3 to the noun `table` then the input has to be reformatted as:

```
3_ t a b l e,
```

where `3_` is the terminal symbol denoting the induced tag cluster 3.

The number of different tags in automatically learned tagset is larger than three, although the training still contains only nouns, verbs and adjectives. This is because even the best unsupervised POS taggers usually learn quite noisy clusters, where one POS category may be split into several different clusters and each cluster may contain a set of words belonging to a mix of different POS categories.

For the random tag baseline we just generate for each word a tag uniformly at random from the set of three tags: $\{0, 1, 2\}$, and reformat the input in a similar way as explained above about the automatically induced tags.

5.3 Evaluation

We evaluate the segmentations using the F1-score of the learned boundaries. The evaluation is type-based (as is also our training), meaning that the segmentation of each word type is calculated into the score only once. This is the simplest evaluation method for morphological segmentation and has been widely used in previous work (Virpioja et al., 2011).

6 Results

We present two sets of results. First we give the F-scores of all evaluated words in each language and then we split the evaluation set into three and evaluate the results for all three POS classes separately.

6.1 General results

The segmentation results are given in Table 2. The first thing to notice is that the models trained with gold-standard POS tags always perform the best. Intuitively this was expected, however, the differences between segmentation F1-scores are in most

[3] The results were obtained from the authors.

	English				Estonian			
	No POS	**Gold**	**Learned**	**Rand**	**No POS**	**Gold**	**Learned**	**Rand**
MorphSeq N	49.5	50.7	52.5	50.0	51.6	56.3	55.0	52.5
MorphSeq V	54.4	59.9	60.7	56.4	46.3	55.9	53.7	47.0
MorphSeq A	50.2	54.6	55.1	51.7	41.1	42.5	44.6	42.7
SubMorph N	61.1	66.9	65.7	61.3	64.6	65.8	64.1	63.8
SubMorph V	67.8	75.4	73.7	70.9	78.9	80.8	75.3	77.8
SubMorph A	61.2	67.0	64.7	60.8	56.6	51.3	51.6	55.5
CollocMorph N	55.0	68.5	65.9	64.3	66.2	67.9	66.8	67.4
CollocMorph V	60.5	75.9	73.4	72.1	68.7	76.0	75.2	79.6
CollocMorph A	54.4	69.1	64.6	62.8	60.0	61.7	55.7	57.6

Table 3: F1-scores of segmentations for different POS classes in English and Estonian using different grammars and settings. **MorphSeq** generates sequences of morphemes, **SubMorph** adds the sub-morphemes, and **CollocMorph** adds the morpheme collocations. **N** denotes nouns, **V** stands for verbs and **A** are adjectives. **No POS** are the models trained without tags, **Gold** uses goldstandard POS tags, **Learned** uses tags learned by an unsupervised POS induction model, and **Rand** uses randomly generated tags.

cases only few percentage points. The only notable exception is English trained with the CollocMorph grammar where the difference with the tag-independent baseline is 14%. However, the baseline score for the CollocMorph grammar in English is much lower than the baseline with the SubMorph grammar, which has a simpler structure. In order to understand why this was the case, we looked at the precision and recall of the CollocMorph grammar results. We found that for the baseline model, the precision is considerably lower than the recall, which means that the results are oversegmented. We always extracted the segmentations from the middle latent level of the CollocMorph grammar and in most cases this gave the best results. However, for the English baseline model, extracting the segmentations from the morpheme collocation level would have given more balanced precision and recall and also a higher F1-score, 60.9%, which would have reduced the difference with the segmentations learned with gold-standard POS tags to 10%.

When the gold-standard POS tags are substituted with the automatically learned tags, the segmentation scores drop as expected. However, in most cases the segmentations are still better than those learned without any tags, although the differences again fall in the range of only few percentage points. In one occasion, namely with the SubMorph grammar in Estonian, the score actually drops by 2% points and with CollocMorph grammar in Estonian the improvement is only about 1%. English segmentations learned with CollocMorph grammar again improve the most over the baseline without tags, gaining over 11% improvement in F1-score.

The last setting we tried used random POS tags. Here we can see that in most cases using random tags helps while in one case—again Estonian SubMorphs—it degrades the segmentation results, leading to lower scores than the baseline without tags. In English, the randomly generated tags always improve the segmentation results over the tag-independent baseline but the results are worse than the segmentations learned with the automatically induced tags. In Estonian, however, for the two more complex grammars, SubMorph and CollocMorph, the randomly generated tags lead to slightly better segmentations than the automatically induced tags. This is a curious result because it suggests that some kind of partitioning of words is helpful for learning better segmentations but that in some cases the resemblance to true POS clustering does not seem that relevant. It could also be that the partitioning of words into nouns, verbs and adjectives only was too coarse for Estonian, which realises many fine-grained morpho-syntactic functions inside each of those POS classes with different suffixes.

6.2 Results of different POS classes

In order to gain more insight into the presented results we also computed F1-scores separately for each of the three POS classes. Those results are given in Table 3. From this table we can see that the segmentation scores are quite different for words with different POS tags. For English, the scores for nouns and adjectives are similar, while the verbs are segmented much more accurately. This is reasonable because usually verbs are much simpler in structure, consisting usually of a stem and a single inflectional suffix, while nouns and adjectives can contain several stems and both derivational and inflectional suffixes. In all cases, segmentations learned with either gold or induced tags are better than segmentations learned with random or no tags at all. CollocMorph is the only grammar where the segmentations learned with random tags improve significantly over the tag-independent baseline. The gap is so large because, as explained above, the precision and recall of the CollocMorph grammar without tags evaluated on the middle grammar level are heavily biased towards recall and the results are oversegmented, while the grammar using randomly generated tags manages to learn segmentations with more balanced precision and recall.

In Estonian, the results are more mixed. For nouns, the only grammar where the POS tags seems to help is the simplest MorphSeq, while with other grammars even specifying gold standard POS tags only leads to minor improvements. Verbs, on the other hand gain quite heavily from tags when using MorphSeq or CollocMorph grammar, while with SubMorph grammar the tag-independent baseline is already very high. Closer inspection revealed that evaluating the CollocMorph grammar on the morpheme collocation level would have given more balanced precision and recall and a tag-independent F-score of 83.2%, which is even higher than the SubMorph 78.9%. Also, evaluating segmentations learned with gold standard tags on that level would have improved the F-score even more up to 89.4%. At the same time, the scores of segmentations learned with both random and induced tags would have dropped. Finally, the scores of the Estonian adjectives are in general the lowest and with both SubMorph and CollocMorph grammar adding the tags in most cases does not give any improvements.

7 Discussion

The main goal of this study was to assess, whether and how much do POS tags help to learn better morphological segmentations. The basis for this question was the intuition that because POS tags and morphological segmentations are linguistically related they should be able to exploit synergies during joint learning. However, the previous work in joint POS induction and morphological segmentation has failed to show the clear gains. Therefore we designed an oracle experiment that uses gold-standard POS tags to measure the upper bound of the gains the POS tags can provide in learning morphological segmentations.

On English, using gold-standard POS tags helps to gain 3-14% of F1-score depending on the grammar, while in Estonian the gains remain between 0-5%. The accuracy gained from tags varies for different POS classes. Both in Estonian and English verbs seem to benefit the most, which can be explained by the fact that in both languages verbs have the simple structure consisting mostly of a stem and an optional inflectional suffix which informs the POS class. At the same time, nouns and adjectives can also contain different derivational morphemes which can be shared by both POS classes. Also, in Estonian the adjectives must agree with nouns in case and number but the sets of suffixes both word classes use are not completely overlapping, which makes the relations between POS tags and segmentations more complex. Another reason for the difference between gains in English and Estonian can be that Estonian as morphologically more complex language may be able to exploit the capacity of the generative AG model more effectively even without tags. At the same time the morphologically simpler English gains more from adding additional information in the form of POS tags.

In general, the effect of POS tags on the segmentation accuracy is not huge, even when the linguistically correct gold-standard tags are used. One reason here can be that we provided the system with very coarse-grained syntactic tags while morphological suffixes are more closely related to the more fine-grained morpho-syntactic functions. This is especially true in English where for instance different verbal suffixes are almost in one-to-one relation with different morpho-syntactic functions. The situation is probably more complex with morphologically rich languages such as Estonian where there are different inflectional classes, which all express the same morpho-syntactic function with different allomorphic suffixes.

Correct	No POS	Gold	Learned	Rand
condemn_ed	cond_em_n_ed	condemn_ed	condem_n_ed	con_demn_ed
grovell_ing	grovel_ling	gro_vell_ing	gro_vell_ing	gro_velling
catalogue	cat_a_logue	cata_logue	cata_logue	cata_logue
propp_ed	pro_p_p_ed	prop_ped	prop_p_ed	propped
match_es	m_atch_e_s	match_es	match_es	matches
suuna_ga (N)	suu_na_ga	suuna_ga	suuna_ga	suunaga
sammal_t (N)	samm_al_t	samm_alt	samm_alt	samm_alt
pääse_ks (V)	pääs_e_ks	pääse_ks	pääse_ks	pääse_ks
pikkuse_d (A)	pikku_sed	pikku_se_d	pikkuse_d	pikku_sed
kükita_sid (V)	kü_ki_ta_sid	küki_ta_sid	küki_tas_id	kükita_sid

Table 4: Examples of both English and Estonian mostly incorrectly segmented words learned with CollocMorph grammar.

Although using automatically induced tags almost always improves the segmentation results, the gains are in most cases quite small. We assume that the induced tags cannot improve the segmentations more than the gold-standard tags. However, it is not clear whether the accuracy of the induced POS tags themselves affects the segmentations accuracy much. The experiments with the random baseline showed that the POS tags should not be completely random but how large differences in tagging accuracies start affecting the segmentations' quality remains to be studied in future works.

Some examples of segmented words for both English and Estonian are given in Table 4. For those examples, the POS-independent grammar has learned incorrect segmentations. The various POS-dependent grammars are in some cases able to learn correct segmentations, in some cases learn more correct segmentations, but in some cases also learn equally false segmentations. For instance for English, all POS-dependent grammars are able to improve the segmenation of the word *condemned*, but only the grammar informed by gold POS tags gets it exactly right. The word *matches* is segmented correctly by grammars using both gold and induced tags, while the grammar with random tags undersegments. In Estonian for instance, only the grammar using random tags gets the word *kükitasid* right, while all the other grammars oversegment it. On the other hand, the adjective *pikkused* is correctly segmented only by the grammar using automatically learned tags and all the other grammars either oversegment or place the segment boundary in an incorrect location.

8 Conclusion

Morphology is a complex language phenomenon which is related to many different phonological, orthographic, morpho-syntactic and morphotactic aspects. This complexity has the potential to create synergies in a generative model where several aspects of the morphology are learned jointly. However, setting up a joint model that correctly captures the desired regularities is difficult and thus it may be useful to study the synergistic potentials of different components in a more isolated setting.

The experiments in this paper focused on the relations between syntactic tags and concatenative morphological segmentations. We showed that both gold-standard POS tags as well as automatically induced tags can help to improve the morphological segmentations. However, the gains are on average not large—5.3% with gold-standard tags and 3.9% with induced tags. Moreover, deeper analysis by evaluating the segmentations of words from different POS classes separately reveals that in Estonian even the goldstandard POS tags do not affect the segmentations much.

These results suggest that perhaps other relations should be studied of how to use various aspects of morphology to create synergies. For instance, POS tags are clearly related to paradigmatic relations. Also, clustering words according to morphosyntactic function could benefit from using methods developed for learning distributional representations. Finally, it could be helpful to learn morphological structures jointly on both orthographic and phonological level.

References

R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. The CELEX lexical database (release 2).

Taylor Berg-Kirkpatrick, Alexandre B. Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590.

Burcu Can and Suresh Manandhar. 2009. Unsupervised learning of morphology by using syntactic categories. In Francesca Borri, Alessandro Nardi, and Carol Peters, editors, *Working Notes for the CLEF 2009 Workshop*.

Burcu Can. 2011. *Statistical Models for Unsupervised Learning of Morphology and POS Tagging*. Ph.D. thesis, The University of York.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2011. A Bayesian mixture model for part-of-speech induction using multiple features. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 638–647.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):3:1–3:34.

Mathias Creutz and Krister Lindén. 2004. Morpheme segmentation gold standards for finnish and english. Publications in Computer and Information Science Report A77, Helsinki University of Technology.

Tomaž Erjavec. 2004. MULTEXT-East version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1535–1538.

Stella Frank, Frank Keller, and Sharon Goldwater. 2013. Exploring the utility of joint morphological and syntactic learning from child-directed speech. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Dayne Freitag. 2005. Morphology induction from term clusters. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CONLL '05)*.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, pages 459–466.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparametric Bayesian inference: Experiments on unsupervised word segmentation with Adaptor Grammars. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Adaptor Grammars: a framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems 19*, pages 641–648.

Mark Johnson. 2008. Unsupervised word segmentation for Sesotho using Adaptor Grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*.

Mark Johnson. 2010. PCFGs, topic models, Adaptor Grammars and learning topical collocations and the structure of proper names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1157.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. 2010. Simple type-level unsupervised POS tagging. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 853–861.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. 2011. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Natural Language Learning*, pages 1–9.

Kairit Sirts and Tanel Alumäe. 2012. A hierarchical Dirichlet process model for joint part-of-speech and morphology induction. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 407–416.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 1(May):231–242.

David Stallard, Jacob Devlin, Michael Kayser, Yoong Keok Lee, and Regina Barzilay. 2012. Unsupervised morphology rivals supervised morphology for arabic mt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, pages 322–327.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 173–180.

Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90.

Structural Alignment as the Basis to Improve Significant Change Detection in Versioned Sentences

Tan Ping Ping, Karin Verspoor and Timothy Miller
Department of Computing and Information Systems
University of Melbourne
Victoria 3010, Australia.
{pingt@student., karin.verspoor@, tmiller@}unimelb.edu.au

Abstract

Some revisions of documents can change the meaning of passages, while others merely re-phrase or improve style. In a multi-author workflow, assisting readers to assess whether a revision changes meaning or not can be useful in prioritising revision. One challenge in this is how to detect and represent the revision changes in a meaningful way to assist users in assessing the impact of revision changes. This paper explores a segmentation approach which utilises the syntactic context of revisions to support assessment of significant changes. We observe that length of normalised edit distance or Word Error Rate (WER) correlates better to the significance of the revision changes at sentence level compared to general sentence similarity approaches. We show that our proposed method, SAVeS, supports improved analysis of change significance through alignment of segments rather than words. SAVeS can be used as the basis for a computational approach to identify significant revision changes.

1 Introduction

Revision of documents is a common component of the writing process. In this work, we introduce an approach to analysing revisions that will support the identification of significant changes, such that attention can be focused on revisions that impact meaning.

We define a *versioned text* as a text document that has been revised and saved to another version, where the original version is directly available for comparison. An *edit* is defined as change that involves operations such as insertion, deletion or substitution of characters or words within a revised text. We define a *significant change* between versioned texts as a meaning altering change, which goes beyond string edit operations.

Faigley and Witte (1981) proposed a taxonomy to assist in evaluating the effect of revisions on meaning (Figure 1). They identify a range of revision types. On a general scale, they define *surface changes* as edits that improve readability without actually changing the meaning of the text, and *text-base changes* as edits that alter the original meaning of the text. These categories are subdivided. The subcategories for surface changes: *formal changes* includes copy editing operations such as correction in spelling, tense, format, etc., while *meaning preserving changes* includes rephrasing. For text-base changes, *microstructure changes* is meaning altering changes which do not affect the original summary of the text and *macrostructure changes* are major changes which alter the original summary of the text. Although they provided some examples, the definitions are insufficient for computational implementation.

Framed by this taxonomy, we consider significant change to be a macro-structure revision change while a minor meaning change is a microstructure revision. We adopt surface revision change to be no meaning change. Based on one original sentence, we provide examples of how we distinguish between meaning-preserving, microstructure and macro-structure revision changes in Table 1.

While some applications use tools like *diff* or come with 'track changes' capability that highlights changes, readers must manually assess the significance over a change, which can reduce efficiency when the number of revisions increases.

In this paper, we demonstrate empirically that general string similarity approaches have weak correlation to significance in revised sentences. We have conducted a preliminary study on a set of revised software use case specifications (UCS)

Ping Ping Tan, Karin Verspoor and Tim Miller. 2015. Structural Alignment as the Basis to Improve Significant Change Detection in Versioned Sentences . In *Proceedings of Australasian Language Technology Association Workshop*, pages 101−109.

Original Sentence	I paid a hundred dollars for the tickets to take my family to a movie.
Revision Type	**Example of Sentence Revisions**
Meaning preserving	I paid a hundred dollars to take my family to a movie.
Micro-structure	I paid a hundred dollars for the tickets, with popcorn and drinks, to bring my family to a movie.
Macro-structure	We decided to watch movie at home.

Table 1: Examples of sentence revision according to revision types

to provide insight into the identification of significant changes between versioned text documents, with particular focus on how impact of revision changes is assessed. The analysis highlights that an approach that considers the syntactic scope of revisions is required for meaning changes assessment.

We will present our proposed method, structural alignment of versioned sentences, SAVeS that addresses this requirement. We provide a performance comparison to three other word segmentation approaches. The broader aim of this research is to develop a computational approach to automatically identifying significant changes between versions of a text document.

2 Related Works

Research on revision concentrates on detecting edits and aligning sentences between versioned text documents. Considering sentences from the first and last draft of essays, Zhang and Litman (2014; 2015) proposed an automated approach to detect whether a sentence has been edited between these versions. Their proposed method starts with sentence alignment, and then identifies the sequence of edits (i.e., the edit operations of Add, Modify, Delete and Keep) between the two sentences. They further consider automated classification of the reason for a revision (i.e., claim, evidence, rebuttal, etc.), which they hypothesised can help writers to improve their writing. Classifying revisions based on the reasons of revision does not indicate the significance of revision changes. What we are attempting is to represent these revision changes in a meaningful way to assist in assessment of the significance. We concentrate on identification of significant revision changes, or revision changes that have higher impact of meaning change for the purpose of prioritising revision changes, especially in multi-author revision. Nevertheless, the work by Zhang and Litman (2014; 2015) provides insights to revisions from a different perspective.

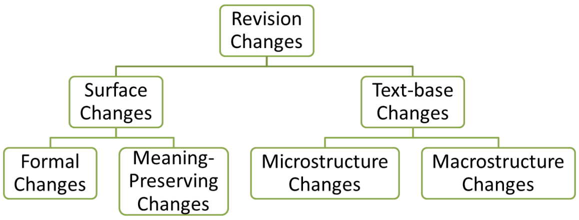


Figure 1: Taxonomy for revision analysis (Faigley and Witte, 1981)

Research has shown that predefined edit categories such as fluency edits (i.e. edits to improve on style and readability) and factual edits (i.e. edits that alter the meaning) in Wikipedia, where revision history data is abundant, can be classified using a supervised approach (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013). The distinction of the edits can be linked to Faigley and Witte's (1981) taxonomy: fluency edits to surface changes and factual edits to text-base changes. Supervised classification would be difficult to apply to other types of revised documents, due to more limited training data in most domain-specific contexts. They too did not consider the significance of edits.

As our task is to align words between versioned sentences to assist in identification of significant changes between versioned texts, it is important to consider the semantics of sentences. Lee et. al. (2014) reviewed the limitations of information retrieval methods (i.e., the Boolean model, the vector space model and the statistical probability model) that calculate the similarity of natural language sentences, but did not consider the meaning of the sentences. Their proposal was to use link grammar to measure similarity based on grammatical structures, combined with the use of an ontology to measure the similarity of the meaning. Their method was shown to be effective for the problem of paraphrase. Paraphrase addresses detecting alternative ways of conveying the same information (Ibrahim et al., 2003) and we observe paraphrase problem as a subset to our task because sentence re-phrasing is part of revision. However, the paraphrase problem effectively try to normalize away differences, while versioned sentences analysis focuses more directly on evaluating the meaning impact of differences.

3 Dataset

The dataset that we study is a set of revised software requirements documents, the Orthopedic Workstation (OWS) Use Case Specifications (UCS) for Pre-Operative Planning for the Hip. We were provided with two versions, version 0.9 (original version, *O*) and version 1.0 (revised version, *R*). Version 1.0 has been implemented as software in a local hospital. Similar to most use case specification documents, the flow of software events, pre- and post-conditions as well as the list of glossary terms are available. The list of glossary terms contains 27 terms with 11 terms having more than one word.

A version that is created immediately following a previous version results in back-to-back versions; these tend to have high similarity to each other. Our dataset consists of back-to-back versions; previous works concentrate on the first and last drafts (Hashemi and Schunn, 2014) (Zhang and Litman, 2014). Therefore in this dataset, we observe more versioned sentences with minor edits that change the meaning substantially (Table 2). Such minor edits are more challenging to determine the significance, from a semantic perspective. These minor edits can be so specific that particular domain knowledge is required to comprehend the changes. We observe 23 pairs of versioned sentences, other than addition and deletion of sentences within this dataset.

4 Introspective Assessment of Revisions

In addition to the summary approach as defined by Faigley and Witte (1981), another approach to distinguish between macro- and micro-structure changes is to determine whether the concepts involved in a particular change affect the reading of other parts of the text. Their definitions are conceptual, for example, they use the notion of a 'gist' to distinguish micro- and macro-structure, but offer no concrete definition of this, such as whether the length of the summary is important, or how much reading of the other parts of the text is influences the summary. Thus, they are not directly suitable as a computational definition. Based on the example in Table 1, we argue that for most cases, micro- and macro-structure can be differentiated without reading the surrounding text, beyond the revised sentences. As our broader objective is to develop a computational method, we conduct our introspective assessment starting at the sentence level, where Zhang and Litman (2014) have demonstrated to work computationally.

Original Sentence, S_O	**Revised Sentence,** S_R
Store X-ray with Current Patient Information.	Store OWS X-ray as Annotated X-ray with Current Patient Record.
Calculate Offset of Non-Destroyed Hip.	Calculate Offset of Normal (Contra-lateral) Hip.
Select material for Insert.	Select material, internal diameter, and other attributes e.g. low profile, extended rim of Insert.

Table 2: Examples of Versioned Sentence Pairs

We observe that changes can be divided into the following three categories:

- **No change:** A pair of sentences which are identical between the versioned texts.
- **Local change:** A change (i.e. word or words added, deleted or modified) where the impact is confined to a pair of versioned sentences.
- **Global change:** A sentence (i.e. added or deleted) where the impact of change is beyond that sentence, for example, at the paragraph or document level.

We will show examples of local changes by considering the first sentence pair in Table 2. A *diff* identify the insertion of "OWS" and "as", "Annotated" and "X-ray", followed by substitution of "Information" to "Record". Based on these edits, readers can roughly estimate words that have changed but cannot assess how much of the meaning has changed. Readers will note that "X-ray" is changed to "OWS X-ray", "as Annotated X-ray" is added and "Patient Information" is substituted with "Patient Record". Readers can only deduce whether the change has any impact when they compare the two versions. "OWS" is the acronym of the system. Although both "OWS X-ray" and "Annotated X-ray" require auxiliary knowledge to identify and understand the changes, the assessment of the impact of the changes is confined within these two sentences or the text surrounding the edits but still within the two sentences. These are examples of *local changes*.

The edit operations observed correspond to the primitive edit operations identified by (Faigley and Witte, 1981; Zhang and Litman, 2014). In our data, there is a minimum of one edit per sentence pair and a maximum of three edits between the pairs. An edit itself can consist of one or multiple words. Substitution and deletion of words and sentences do occur, but a large number of the edits involve adding words to the later version. Most additions provide more clarification; 16 out of the local (i.e., word) additions contribute to either minor or major meaning change. Thus, local changes can be either significant or not.

Global changes have no matching or similar sentence between the two versions, unlike the other two changes. Most of the assessment of the impact of global changes is based on the preceding sentences, which can be either a revised sentence or an unchanged sentence. Even though we do not work on global changes in this paper, we provide an example differentiating local and global changes (Table 3).

Original, O	**Revised, R**
Label pathology on X-ray.	Label pathology on Annotated X-ray. Predefined Labels includes suggestions.
Local changes	'X-ray' to 'Annotated X-ray'
Global Change	'Predefined Labels includes suggestions.'

Table 3: Example of Local and Global Changes

Our introspection highlights three main things, which serve as motivation for this work:

- The need for local and global changes to be differentiated, before micro- and macro-structure differentiation.
- The way readers assess the impact of change depends upon both syntactic and semantic understanding of the changes.
- The words surrounding the edits are useful for assessment of impact of revision changes.

5 Structural Alignment of Versioned Sentences

Chomsky (2002) suggested that "structure of language has certain interesting implications for semantics study". The idea of using sentence structure in natural language specification to describe program input data has been proposed by Lei (2013). Based on this notion, and the understanding of how local changes are assessed through our introspective study, we present a method to group words into segments. Specifically, we propose to use the sentence structure, corresponding to the syntactic context of the edited words, to assist in alignment of versioned sentences. Then we make use of these segments in assessing the impact of revision changes.

Our proposed Structural Alignment for Versioned Sentences (SAVeS) method starts by performing tokenization, where each word is treated as single token, for each of the sentences, producing T_{S_O} and T_{S_R}. Tokens that are the same between T_{S_O} and T_{S_R} are aligned, leaving the edited words from each sentence, E_{S_O} and E_{S_R}. In a separate process, each of the sentences serves as input to a syntactic parser, producing individual parse trees, PT_{S_O} and PT_{S_R}. SAVeS matches each of the edited words to the leaves of the parse trees, then extracts the head of the noun phrase for each edited word. The tokens in T_{S_O} and T_{S_R} are updated according to the grouped words (i.e. noun phrase of the edited words), producing T'_{S_O} and T'_{S_R}. Words that are not part of an edited phrase continue to be treated as individual tokens. Using S_O from the first example in Table 2, we provide a sample of how SAVeS captures the context of the edited word (in this case: 'information') in Figure 2 and the full SAVeS algorithm appears in Table 4.

SAVeS uses general sentence structure, therefore, is applicable to different types of phrases. In this dataset, majority of phrases are noun phrases. As a preliminary, we work on noun phrasest.

Algorithm	Structural Alignment of Versioned Sentences
Input	Versioned Sentences: Original Sentence, S_O and Revised Sentence, S_R
Output	Word Error Rate, WER
	POS - Part Of Speech NP - Noun Phrase

```
For each sentence,
    $T_S$ = Tokenise each word in the sentence
End For
Align the words that are the same between $T_{S_O}$ and $T_{S_R}$,
    Extract the edited words for each of the sentence
For each of the sentence,
    $PT_S$ = Constituency-based parse tree
    For each of the edited word
        For each leaf
            If leaf value = edited word,
                While node POS not equal to NP,
                    Get the POS of the parent of node
                End While
                Extract the NP
            End If
        End For
    End For
End For
For each of the extracted phrases
    $T'_S$ = Group the tokens based on the extracted phrase
End For
```

Table 4: Algorithm for Structural Alignment of Versioned Sentences

6 Experimental Setup

6.1 Measuring Revisions

The experiments measure revision changes at sentence and word segmentation level. String similarity is used to measure the surface similarity of two sentences, while semantic similarity measure whether two sentences have the same meaning. Therefore, we consider pairwise string and semantic similarity between sentences; pairs that are more different are considered to have more significant changes.

Given two strings, x and y, the edit distance between x and y is the minimum editing path to transform x to y, where edit path covers operations like substitution, insertion and deletion of word or character, taking into consideration of word order. Our work on revision sentences observes the transformation from the original sentence, S_O to the revised sentence, S_R. The length of the sentences can vary. Hence, we consider the length of normalised edit distance or Word Error Rate (WER) (Equation 1). WER is an automatic evaluation metric commonly used in machine translation to observe how far off the system output is from a target translation. In our case, it is used to automatically measures how different S_O and S_R is.

$$WER(S_O, S_R) = \frac{W(P)}{maximum\_length(S_O, S_R)} \tag{1}$$

Where:

P is minimum edit distance between S_O and S_R,

W(P) is the sum of the edit operations of P, where weight is added for edit operation involving word in the glossary for the weighted glossary experiment.

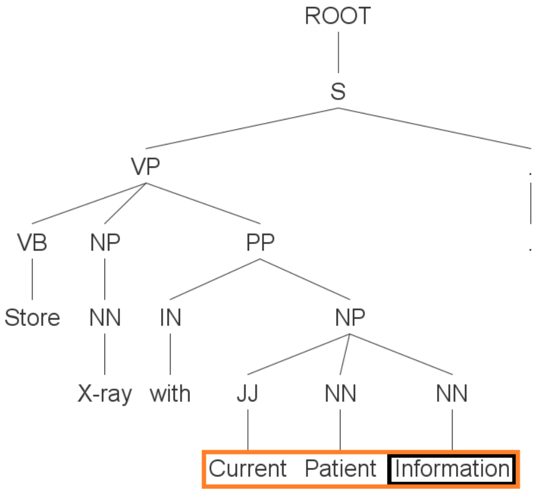


Figure 2: Example how SAVeS capture the context surrounding the edited word

6.2 Annotation

Before we can consider a suitable measurement for revision changes between versioned sentences, manual intuitive annotation is performed by an annotator, with review from one other. The versioned sentences are annotated based on significance of the changes, framed by Faigley and Witte's (1981) revision analysis taxonomy. We compared the original sentence, S_O, to the revised sentence, S_R, and for each sentence pair determined whether there is a meaning change. We first differentiate between surface and text-base revision changes. If the revision is a text-base change, we further distinguish between the micro- or macro-structure levels. The versioned sentences can have more than one local change; therefore, we annotate the sentence pair as *non significant, minor* and *significant* change based on the most significant change for that sentence pair.

Each of the measurements stated in Section 6.1 is plotted against this human annotation of significance, followed by the calculation of correlation coefficient, r values between the labels. If r value closer to 1, the measurement correlates better with the significance, while opposite correlation is observed for negative r value. When r value is closer to 0, weak correlation between the variables.

6.3 Similarities and Significant Revisions

The versioned sentence pairs serve as the input to the similarity approaches, and the output is the similarity values for each of the sentence pairs. For string similarity measurement, we used Jaro-Winkler proximity (Cohen et al., 2003). Automatic machine translation evaluation metrics, which normally integrate with linguistics knowledge, is used to measure how semantically similar between the translation output of a system to the parallel corpus without human judgement. This approach is also used for paraphrase evaluation (Madnani et al., 2012). For semantic similarity, we adopted one of the metrics, Tesla (Liu et al., 2010), which is linked to WordNet as our semantic similarity measurement between versioned sentences.

6.4 Word Segmentation impact on revision

For the task of word segmentation, we consider four scenarios. In each case, the alignment is computed using edit distance based on the relevant segmentation (considering insertions, deletions, and substitutions of *segments*). The word error rate (WER) or the length of normalised edit distance (Equation 1) is computed on the basis of this alignment.

- **Baseline:** We use the standard approach of treating a single word as a single token. In the alignment of S_O and S_R, matching tokens are aligned. We use this as the baseline approach.

- **Glossary:** In this approach, we consider changes in domain-specific terminology are more likely to impact the meaning of the sentence. Instead of just tokenizing on the individual terms as separate tokens, the terms that exist in the glossary terms are grouped together as a token, while the other words remained as single tokens.

- **Weighted Glossary:** Here, we consider that edited words in the versioned sentences that exist in the glossary list may have more importance. We added weights to these edited words in the edit distance calculation to emphasize their importance in aligning the glossary terms. In this scenario, similar to the second scenario, the glossary is used to guide tokenization, with addition that penalizes edits involving these glossary-based tokens more heavily. As there is no previous work on the optimal weight to use for aligning versioned sentences, we experimented with a weight value of $+2$.

- **SAVeS:** SAVeS is implemented based on the algorithm in Figure 4. The updated tokens are

Approach	r
String Similarity	-0.34
Semantic Similarity	-0.59
Tokenization approaches:	
Baseline	0.63
Glossary Terms	0.66
Weighted Glossary Terms	0.68
SAVeS	0.58

Table 5: Correlation coefficient (r) values between similarity measurement and significant changes, using various approaches to similarity assessment.

re-aligned based on the noun phrases. The Stanford parser (Klein and Manning, 2003) we used produced parse trees with minor errors in some sentences. To eliminate issues in the results related to the incorrect parsing, we manually corrected errors in the parse trees, thus assuming the existence of a 'perfect' parser.

7 Results and Discussion

Table 5 shows that semantic similarity has a stronger negative correlation to significant changes when compared to string similarity but the baseline approach of single word token alignment correlates better to significant changes. This result shows that semantic similarity could be used to filter out non-significant revised sentences before further evaluation of micro- and macro-structure assessment.

Using the weighted glossary term tokenization approach, the WER correlates best with the significance at sentence level, compared to the other tested approaches. A domain specific dataset clearly benefits from specific knowledge of terminology. However, we still do not understand the most appropriate weights to use. A more detailed study is required to fully determine the optimal weights for integrating the glossary to assist in producing an analysis of the impact of revision changes.

The human annotation of significance is based on the highest significance between the versioned sentence pair. Although for cases where there is more than one changes between the versioned sentence pairs, using WER evaluation cannot pinpoint which among the changes in that sentence pair is indeed significant.

Table 6 presents an analysis of the effect of different tokenization approaches and WER, based on the first example in Table 2, where the glossary terms are 'annotated x-ray' and 'patient information'. When we examine the changes after alignment more closely, the baseline approach outputs the edits between the two sentences without much indication of meaning changes. The glossary terms tokenization approach is able to treat 'annotated x-ray' as a single insertion and although 'patient record' appears as a segment but aligns to 'patient' it is not reflective of the meaning change, instead for this change, 'patient record' is substituted to 'patient information' should be a better representation to evaluate the meaning change.

Weighting glossary terms emphasizes the changes introduced by a shift in core terminology, the addition of 'annotated x-ray'. SAVeS identifies the main segments: 'annotated x-ray', which we can deduce as insertion of a noun phrase, 'x-ray' is substituted with 'ows x-ray', which we can be deduced is a type of X-ray and 'current patient information' is substituted with 'current patient record' which shows us, this is a possible meaning preserving change.

When we compare the relationship between these different tokenization approaches and the WER, we see that the weighted glossary term tokenization approach reflects a larger change between the sentences (i.e., WER = 0.78) compared to other tokenization approaches.

We examined the impact of the different tokenization approaches on the WER, according to the manually assigned significance category (Table 7). For the significance categories of *None* and *Minor*, the alignment using SAVeS measures less change (i.e. substitution, insertion and deletion) as compared to other tokenization approaches.

Consider the second example in Table 2. SAVeS extracted phrases that contain the edited words and aligned them, rather than individual words: the full phrase 'non-destroyed hip' is aligned by the phrase 'normal (contra-lateral) hip'. In this case, the WER for single word single token alignment (i.e., baseline) is 0.33 while SAVeS produces 0.25. SAVeS reflects that the scope of the edits is limited to one (syntactically bounded) portion of the sentence.

SAVeS highlights meaning changes by supplying the information that the full phrase 'non-

Tokenization Approach	Tokens	WER	Changes Detected
Baseline	S_O = {store, ows, x-ray, as, annotated, x-ray, with, current, patient, record} S_R = {store, x-ray, with, current, patient, information}	0.5	*insertion*: ’ows’, ’as’, ’annotated’, ’x-ray’ *substitution*: ’record’ to ’information’
Glossary Terms	S_O = {store, ows, x-ray, as, annotated x-ray, with, current, patient record} S_R = {store, x-ray, with, current, patient, information}	0.56	*insertion*: ’ows’, ’as’, ’annotated x-ray’ *substitution*: ’patient’ to ’patient record’ *deletion*: ’information’
Weighted Glossary Terms	S_O = {store, ows, x-ray, as, annotated x-ray, with, current, patient record} S_R = {store, x-ray, with, current, patient, information}	0.78	*insertion*: ’ows’, ’as’, ’annotated x-ray’ (weight: +4) *substitution*: ’patient’ to ’patient record’ (weight: +4) *deletion*: ’information’
SAVeS	S_O = {store, ows x-ray, as, annotated x-ray, with, current patient record} S_R = {store, x-ray, with, current patient information}	0.67	*insertion*: ’as’, ’annotated x-ray’ *substitution*: ’x-ray’ to ’ows x-ray’, ’current patient information’ to ’current patient record’

Table 6: An example of tokenization effect and WER.

destroyed hip’ is substituted by ‘normal (contralateral) hip’. Deduction of the impact can only be made if this substitution is analysed in more depth. Observe that the rightmost noun in the phrase (i.e., ‘hip’; the syntactic and semantic head of the phrase) did not change; this too may have implications for the assessment of meaning. A few more other examples of the effect of SAVeS through analysis of the tokens alignment can be considered: ‘surgeon authentication’ is aligned to ‘authentication’ or ‘labelled image’ is aligned to ‘labelled annotated x-ray’ where other tokenization approaches cannot chunk and align these changes. The advantage of SAVeS over the glossary terms approach is that not all of the terms exist in the glossary list. Using the sentence syntactic structure, SAVeS is applicable to any sentence.

For the case of significant revision changes, the changes are small irrespective of the tokenization approach. This is due to the nature of our dataset; back-to-back versions. The small average WER across the category of significant changes shows that edits alone are insufficient to bring out the semantics of the changes.

Significance	SAVeS	BL	Gl	W-Gl
None	0.24	0.25	0.26	0.33
Minor	0.35	0.45	0.46	0.49
Significant	0.19	0.24	0.25	0.25

Table 7: The average WER by revision significance, based on each different tokenization approach (BL=baseline, Gl=glossary, W-Gl=weighted glossary)

.

We hypothesize that phrases will provide a better representation for meaning change analysis between versioned sentences than individual tokens, and further suggest that measuring edits at the phrasal level will lead to an improvement in our ability to computationally determine the significance of changes.

In a multi-author environment, the current tools only provide the edits of the revision but SAVeS indicates which of the noun phrases have changed. We hypothesise that this form of indicator is more useful to authors.

8 Conclusion

Our introspective assessment of revision changes in versioned use case specifications revealed that changes can be categorised into local and global changes, and that there exist versioned sentences which can be superficially similar and yet reflect substantial differences in meaning. In order to make direct comparison between changes for the purpose of assessment, we need to consider the context of a change. We empirically show that alignment of words between versioned sentences using word error rate correlates better to significance of a revision. In this paper, we have explored several approaches to aligning versioned sentences in this context. Our analysis of the alignment shows that incorporating structural information of the text affected by an edit is useful for taking into consideration the scope of an edit in its sentential context. We further demonstrate that similarity approaches are insufficient for our task.

We speculate that a phrasal representation of revisions will also be better for human readability of edits during manual assessment of the significance of changes, and plan to assess this in future work. This is a preliminary study and we plan to consider other kinds of versioned documents.

References

Amit Bronner and Christof Monz. 2012. User edits classification using document revision histories. In *EACL*, pages 356–366. Assoc. for Computational Linguistics.

Noam Chomsky. 2002. *Syntactic structures*. Walter de Gruyter.

WW Cohen, P Ravikumar, and S Fienberg. 2003. Secondstring: An open source java toolkit of approximate string-matching techniques. *Project web page: http://secondstring. sourceforge. net*.

Johannes Daxenberger and Iryna Gurevych. 2013. Automatically classifying edit categories in wikipedia revisions. In *EMNLP*, pages 578–589.

Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College composition and communication*, pages 400–414.

Homa B Hashemi and Christian D Schunn. 2014. A tool for summarizing students changes across drafts. In *Intelligent Tutoring Systems*, pages 679–682. Springer.

Ali Ibrahim, Boris Katz, and Jimmy Lin. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second int'l workshop on Paraphrasing*, pages 57–64. Assoc. for Computational Linguistics.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *ACL*, pages 423–430. Assoc. for Computational Linguistics.

Ming Che Lee, Jia Wei Chang, and Tung Cheng Hsieh. 2014. A grammar-based semantic similarity algorithm for natural language sentences. *The Scientific World Journal*, 2014.

Tao Lei, Fan Long, Regina Barzilay, and Martin C Rinard. 2013. From natural language specifications to program input parsers. Assoc. for Computational Linguistics.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. Tesla: Translation evaluation of sentences with linear-programming-based analysis. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics*, WMT'10, pages 354–359, Stroudsburg, PA, USA. Assoc. for Computational Linguistics.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *NAACL/HLT*, pages 182–190. Assoc. for Computational Linguistics.

Fan Zhang and Diane Litman. 2014. Sentence-level rewriting detection. *ACL 2014*, page 149.

Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143. Assoc. for Computational Linguistics.

Short papers

More Efficient Topic Modelling Through a Noun Only Approach

Fiona Martin
Department of Computing,
Macquarie University, NSW, 2109,
Australia
fiona.martin@students.mq.edu.au

Mark Johnson
Department of Computing,
Macquarie University, NSW, 2109,
Australia
mark.johnson@mq.edu.au

Abstract

This study compared three topic models trained on three versions of a news corpus. The first model was generated from the raw news corpus, the second was generated from the lemmatised version of the news corpus, and the third model was generated from the lemmatised news corpus reduced to nouns only. We found that the removing all words except nouns improved the topics' semantic coherence. Using the measures developed by Lau et al (2014), the average observed topic coherence improved 6% and the average word intrusion detection improved 8% for the noun only corpus, compared to modelling the raw corpus. Similar improvements on these measures were obtained by simply lemmatising the news corpus, however, the model training times are faster when reducing the articles to the nouns only.

1 Introduction

A challenge when analysing a large collection of text documents is to efficiently summarise the multitude of themes within that collection, and to identify and organise the documents into particular themes. Document collections such as a newspaper corpus contain a wide variety of themes or topics, with each individual article referencing only a very small subset of those topics. Such topics may be broad and coarse grained, such as *politics*, *finance* or *sport*. Alternatively, topics may be more specific, such as articles related to earthquakes in southern California, or to Napa Valley wineries.

Topic modelling is one way to examine the themes in large document collections. Topic modelling considers documents to be a mixture of latent topics. A more formal definition of topics, as provided by Blei (2012), is that a topic is a multinomial distribution over a fixed vocabulary. One of the most prominent algorithms for topic modelling is the Latent Dirichlet Allocation (LDA) algorithm, developed by Blei et al. (2003). Typically the most frequent function words are excluded prior to topic modelling with LDA (termed *stop word removal*). The topics then generated by LDA can be a mixture of nouns, verbs, adjectives, adverbs and any function words not previously excluded. The LDA algorithm treats all word tokens as having equal importance.

It is common to examine the most frequent words associated with the topic, to determine if these words together suggest a particular theme. For example, a topic with the most frequent words {*water plant tree garden flower fruit valley drought*} suggests a possible label of "garden", whereas a topic of {*art good house room style work fashion draw*} seems to combine multiple themes, and is harder to label. Manually assigning a meaning to a topic (e.g. "gardening") is easier for a reviewer if the most frequent words in the topic are semantically coherent. One issue identified with topic modelling is that it can generate 'junk' topics (Mimno et al., 2011), that is, topics lacking coherence (as in the second example above). Such topics are either ambiguous or have no interpretable theme.

While in some instances there may be interest in examining adjectives (say for sentiment analysis), or verbs (if seeking to identify change, for example), often interest centres around entities such as people, places, organisations and events. For articles drawn from all sections of a newspaper (for example, *Sport*, *Business*, *Lifestyle*, *Drive* and so on), it may be useful to organise articles ignoring their section of origin, and instead focus on the subjects of each article, that is, the people, places, organisations and events (e.g. *earthquake* or *Election*). Such information is typically represented in the articles' nouns.

This study builds on the work of Griffiths et al.

Fiona Martin and Mark Johnson. 2015. More Efficient Topic Modelling Through a Noun Only Approach . In *Proceedings of Australasian Language Technology Association Workshop*, pages 111−115.

(2005), Jiang (2009) and Darling et al. (2012), where topics were generated for specific parts of speech. The novelty in this current study is that it is concerned solely with noun topics, and reduces the corpus to nouns prior to topic modelling. As a news corpus tends have a broad and varied vocabulary, that can be time consuming to topic model, limiting articles to only the nouns also offers the advantage of reducing the size of the vocabulary to be modelled.

The question of interest in this current study was whether reducing a news corpus to nouns only would efficiently produce topics that implied coherent themes, which, in turn, may offer more meaningful document clusters. The measures of interest were topic coherence and the time taken to generate the topic model. Previous work by Lau et al. (2014) suggests that lemmatising a corpus improves topic coherence. This study sought to replicate that finding, and then examine if further improvement occurs by limiting the corpus to nouns. The news corpus and the tools applied to that corpus are detailed in the next section. Section 3 provides the results of the topic coherence evaluations, and Section 4 discusses these results in relation to the goal of efficiently generating coherent topics.

2 Data and Methods

2.1 Data and Pre-Processing

Topic models were generated based on a 1991 set of San Jose Mercury News (SJMN) articles, from the Tipster corpus (Harman & Liberman, 1993). The articles in this corpus are in a standard SGML format. The SGML tags of interest were the <HEADLINE>, <LEADPARA>and <TEXT>, where the lead paragraph of the article has been separated from the main text of the article. The SJMN corpus consisted of 90,257 articles, containing 35.8 million words. Part-of-speech (POS) tagging identified 12.9 million nouns, which is just over 36% of the total corpus. The POS tagging meant a single token such as '(text)' was split into three tokens: '(', 'text',')'. Such splits resulted in the lemmatised set of articles being larger, with over 36.2 million tokens. As this split would be done by the topic modelling tool anyway, it made no material difference to the topics generated, but it did increase the number of tokens fed to the topic modeller, slowing the topic generation.

The news articles were pre-processed by part-of-speech (POS) tagging and each word token was lemmatised. POS tagging was done using the Stanford Log-linear Part-of-Speech tagger (StanfordPOS) (Toutanova et al., 2003), v3.3.1 (2014-01-04), using the *wsj-0-18-bidirectional-distsim.tagger* model. The Stanford POS tagger is a maximum-entropy (CMM) part-of-speech (POS) tagger, which assigns Penn Treebank POS tags to word tokens. Following the finding of Lau et al. (2014) that lemmatisation aided topic coherence, the news articles were lemmatised for the second and third versions of the corpus (but not the first set of articles, to be referred to as the *Original* version of the corpus). Lemmatisation was performed using the *morphy* software from NLTK[1], version 2.0.4, and was applied using the POS tag identified for each word. The *morphy* function reduced words to their base form, such as changing 'leveraged' to 'leverage', and 'mice' to 'mouse'. A Python script was used to create a version of the SJMN articles that contained only tokens tagged with the Penn Treebank noun type tags.

Three distinct versions of the articles were formed, to generate three separate series of topic models. The first version was the complete, original SJMN articles. The second version was a lemmatised set of SJMN articles. The third version was a lemmatised set of SJMN articles, reduced to only nouns. Punctuation was removed from the text in all three versions of the news corpus.

2.2 Topic Modelling

Topic modelling was performed using the Mallet software from the University of Massachusetts Amherst (McCallum, 2002). The Mallet software was run to generate topics using the Latent Dirichlet Allocation (LDA) algorithm, configured to convert all text to lowercase, to model individual features (not n-grams), and to remove words predefined in the Mallet English stop-word list prior to topic modelling. The default settings were used for the optimise-interval hyperparameter (20) and the Gibbs sampling iterations (1,000). The Mallet software uses a random seed, so the resulting topics can vary between models even when generated using the exactly the same settings and corpus. It is expected that, on balance, dominant topics should re-occur each time the topics are generated, but the nature of such unsupervised learning means that this may not al-

[1] http://www.nltk.org/howto/wordnet.html

ways be the case. To account for such variation, topic models were generated ten times for each set of the news articles, and scores averaged across those ten runs.

The Mallet software requires the number of topics to be specified in advance. As there is not yet an agreed best method for determining the number of topics, this study generated separate sets of 20, 50, 100, 200 and 500 topics. All showed similar patterns between the three data sets. The 200 topics produced the highest topic coherence, as assessed by the measures described in the next section, and for brevity, only the results of the 200 topic runs are reported in this paper.

Table 1: *Average Topic Coherence Measures*

Version	Mean (SD)	Median	Range
1. Original	0.162 (0.087)	0.160	0-0.52
2. Lemmatised	0.170 (0.086)	0.165	0-0.49
3. Nouns Only	0.172 (0.081)	0.170	0-0.49

For each version of the articles, OC scores were averaged across the 200 topics, across the ten topic models (n=2,000).

Table 2: *Number of Low Coherence Topics*

Version	OC <0.1	OC = 0
1. Original	409 (20%)	16 (8%)
2. Lemmatised	346 (17%)	9 (5%)
3. Nouns Only	305 (15%)	1 (1%)

Counts are across the ten models of 200 topics (i.e. n=2,000). The figures in brackets are a percent of the 2000 total topics, for each article set.

2.3 Topic Evaluation

The study by Lau et al. (2014)[2] produced two measures found to be well correlated with human evaluations of topic coherence, and those two measures were used in this current study. The first was an observed coherence (OC) measure, that was configured to use normalised point-wise mutual information (NPMI) to determine how frequently words co-occur in a corpus, and then use this to measure the coherence of the top ten most frequent words in each topic. An NPMI OC score closer to 1 reflected greater co-occurrence, whereas a score of 0 indicated the words were independent.

The second measure was an automated word intrusion detection (WI) task. This task required an intruder word to be inserted into a random location in each topic. The intruder words needed to be words common to the corpus, but not related to the themes in the individual topic. The WI software used the word co-occurrence statistics from the reference corpus to choose which word was most likely to be the intruder. The WI software rated accuracy as either detected (1) or not detected (0).

The proportion of topics where the WI software automatically detected the intruder word was calculated per model via a Python script. This result was expressed as a proportion between 0 and 1, with a value of 0.5 indicating that only half of the intruder words were detected across all (200) topics. A proportion of 1 would indicate all intruder words detected, and 0 indicated no intruder words were detected in any topics. The San Jose Mercury corpus was used as the reference corpus for calculating the baseline co-occurrence.

A final check determined the percentage of nouns in the top 19 most frequent words for each topic. This check was done only for topics generated from the original corpus. To be counted as a noun, a word must have been POS tagged as a noun somewhere in the corpus (for example, “burden” might appear as both a verb and a noun at different places in the corpus, but will be counted as a noun for this statistic).

3 Results

The NPMI Observed Coherence (OC) proportions and the Word Intrusion (WI) detection percentages are shown in Table 1 and 3, respectively. These figures suggest an improvement in topic coherence in the second and third models. Table 2 indicates that all three article sets produced substantial numbers of topics with very low coherence scores. The *Nouns Only* articles produced the least number of low and zero OC coherence topics, suggesting lower numbers of ‘junk’ topics. Additionally, a review of the topics generated from the original (unaltered) article set indicated a clear predominance of nouns, with over 99% of the 19 most frequent words being nouns, for each of the 200 topics.

It must be noted that the OC scores suggest it was a different set of 200 topics generated each of the ten times topic modelling was performed on the same versions of the articles. For a given version of the articles, none of the ten models produced the same average OC scores as another model on that article set. For example, of the ten models for the *Lemmatised* articles, the mean OC scores ranged between 0.1679 and 0.1744, but no two

[2] The software used in the evaluations was downloaded from https://github.com/jhlau/topic_interpretability, on the 1 May 2014.

Table 3: *Average Word Intrusion Detection*

Version	Mean (SD)	Median	Range
1. Original	0.80 (0.03)	0.79	0.77-0.86
2. Lemmatised	0.88 (0.02)	0.89	0.84-0.90
3. Nouns Only	0.87 (0.03)	0.87	0.83-0.91

Average WI scores were calculated for each of the ten 200 topic models, and the averages of these ten are shown here, for each version of the articles topic modelled (n=200).

Table 4: *Average Time to Generate 200 Topics*

	Time (mins)
Version	Mean (SD)
1. Original	92 (1)
2. Lemmatised	104 (2)
3. Nouns Only	75 (3)

were the same. Minimum, maximum and median OC scores showed similar differences across the ten models. These differences indicate that the generated topics were different in each of the ten models generated for a given article set.

Finally, Table 4 shows that the nouns only corpus was faster to topic model than the other two versions of the news corpus. Part-of-speech tagging the articles took, on average, less than one second per article. Memory restrictions encountered with the part-of-speech tagger meant the articles had to be tagged in parallel sets, rather than tagging the complete corpus at once.

4 Discussion

For the two measures evaluated in this study, reducing the SJMN news corpus to only nouns produced equivalent or improved topic semantic coherence, compared to topic modelling the original news articles. Interestingly, even when the original articles contained all words (apart from the stop words), topic modelling still favoured nouns as the most frequent words in the topics. This suggests that reducing the articles to only nouns may be advantageous in that it may remove extra vocabulary items that would not typically be ranked highly among the most frequent words of a topic anyway. The results of this study suggest that for topic coherence, lemmatising the articles could be the most important factor. However, lemmatising alone does not reduce the time taken to generate the topic model.

Drawing conclusions about any performance impacts is more problematic due to the separate, unintegrated nature of the POS tagging and topic modelling used in this study. There was additional time taken for intermediate file operations that could be eliminated in an integrated process (e.g. piping output between tagging and modelling). Future research could look to integrating the POS tagger and the topic model to gain the best efficiency advantage.

The measures of topic coherence used here are based on whether the top ten most frequent words for a topic are words that commonly co-occur. It does not validate whether these words represent a topic which truly reflects one of the top 200 most frequent themes across articles in the corpus. The substantial variability in both the topic coherence and the word intrusion detection indicate it was not the same 200 topics in each of the ten models generated, for each set of articles. This was confirmed by manual reviews of the topics generated, for each of the three sets of the articles. This variability also occurred when more topics were generated (i.e. 500 topic models) and less topics (i.e. 20, 50, 100 topic models). Though variability is not unexpected in an unsupervised method such as topic modelling, such variability indicates the topics may be unreliable, and is of concern if the end-user seeks to draw detailed conclusions about a corpus based on a single topic model. For example, if a topic related to earthquakes occurred in one set of topics, then it cannot be guaranteed that if the model is re-generated, that such an earthquake topic will re-occur. Therefore, caution should be applied when using topics to make inferences about a corpus, and all inferences should be cross checked using alternate means.

5 Conclusion and Future Work

This study replicated the findings of Lau et al. (2014) that lemmatising improves topic coherence, on observed coherence and word intrusion measures. Limiting the lemmatised corpus to nouns only retains this coherence advantage, while reducing model generation time. Therefore, this study found that lemmatising and limiting the news corpus to the nouns offers advantages in topic coherence and speed, compared to topic modelling the raw corpus of SJMN articles, or lemmatising alone. While this study considered topic coherence, future work could seek to improve topic reliability (i.e. topic consistency). This may include new measures of topic reliability, and optimising the number of topics that can be reliably generated for a given corpus.

Acknowledgments

This research is funded by the Capital Markets Cooperative Research Centre.

References

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Blei, D. M., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Darling, W. M., Paul, M. J., & Song, F. (2012). Unsupervised part-of-speech tagging in noisy and esoteric domains with a syntactic-semantic bayesian hmm. In *Proceedings of the Workshop on Semantic Analysis in Social Media* (pp. 1–9). Stroudsburg, PA, USA: The Association for Computational Linguistics.

Griffiths, T. L., Steyvers, M., Blei, D. M., & Tenenbaum, J. B. (2005). Integrating topics and syntax. In Saul, L., Weiss, Y., & Bottou, L. (Eds.), *Advances in Neural Information Processing Systems 17*, (pp. 537–544). MIT Press.

Harman, D. & Liberman, M. (1993). TIPSTER Complete LDC93T3A. DVD. `https://catalog.ldc.upenn.edu/LDC93T3D`.

Jiang, J. (2009). Modeling syntactic structures of topics with a nested HMM-LDA. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining* (pp. 824–829). Washington, DC, USA: IEEE Computer Society.

Lau, J. H., Newman, D., & Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)* (pp. 530–539). Gothenburg, Sweden: Association for Computational Linguistics.

McCallum, A. K. (2002). MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (pp. 262–272). Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, (pp. 173–180). The Association for Computational Linguistics.

Improving Topic Coherence with Latent Feature Word Representations in MAP Estimation for Topic Modeling

Dat Quoc Nguyen, Kairit Sirts and **Mark Johnson**
Department of Computing
Macquarie University, Australia
dat.nguyen@students.mq.edu.au, {kairit.sirts, mark.johnson}@mq.edu.au

Abstract

Probabilistic topic models are widely used to discover latent topics in document collections, while latent feature word vectors have been used to obtain high performance in many natural language processing (NLP) tasks. In this paper, we present a new approach by incorporating word vectors to directly optimize the maximum a posteriori (MAP) estimation in a topic model. Preliminary results show that the word vectors induced from the experimental corpus can be used to improve the assignments of topics to words.

Keywords: MAP estimation, LDA, Topic model, Word vectors, Topic coherence

1 Introduction

Topic modeling algorithms, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and related methods (Blei, 2012), are often used to learn a set of latent topics for a corpus of documents and to infer document-to-topic and topic-to-word distributions from the co-occurrence of words within the documents (Wallach, 2006; Blei and McAuliffe, 2008; Wang et al., 2007; Johnson, 2010; Yan et al., 2013; Xie et al., 2015; Yang et al., 2015). With enough training data there is sufficient information in the corpus to accurately estimate the distributions. However, most topic models consider each document as a bag-of-words, i.e. the word order or the window-based local context information is not taken into account.

Topic models have also been constructed using latent features (Salakhutdinov and Hinton, 2009; Srivastava et al., 2013; Cao et al., 2015). Latent feature vectors have been recently successfully exploited for a wide range of NLP tasks (Glorot et al., 2011; Socher et al., 2013; Pennington et al., 2014). Rather than relying solely on word count information as the standard multinomial LDA does, or using only distributed feature representations, as in Salakhutdinov and Hinton (2009), Srivastava et al. (2013) and Cao et al. (2015), Nguyen et al. (2015) integrated pre-trained latent feature word representations containing external information from very large corpora into existing topic models and obtained significant improvements on small document collections and short text datasets. However, their implementation is computationally quite expensive because they have to compute a MAP estimate in each Gibbs sampling iteration.

In this paper, we experiment with MAP estimation using word vectors for LDA. Instead of mixing the Gibbs sampling and MAP estimation, we propose to optimize the MAP estimation of the full model directly. In addition, instead of using the pre-trained word vectors learned on external large corpora, we propose to learn the internal word vectors from the same topic-modeling corpus that we induce the document-to-topic and topic-to-word distributions from. In this manner, we can also handle the words that are not found in the list of the pre-trained word vectors. Furthermore, the internal word vectors can capture various aspects including word order information or local context information in the topic-modeling corpus. Preliminary results show that the internal word vectors can also help to significantly improve the topic-to-word assignments.

2 Related work

LDA (Blei et al., 2003) represents each document d in the document collection D as a mixture $\boldsymbol{\theta}_d$ over T topics, where each topic z is modeled by a probability distribution $\boldsymbol{\phi}_z$ over words in a vocab-

Dat Quoc Nguyen, Kairit Sirts and Mark Johnson. 2015. Improving Topic Coherence with Latent Feature Word Representations in MAP Estimation for Topic Modeling . In *Proceedings of Australasian Language Technology Association Workshop*, pages 116−121.

ulary W. As presented in Figure 1, where α and β are hyper-parameters, the generative process for LDA is described as follows:

$$\boldsymbol{\theta}_d \sim \text{Dir}(\alpha) \qquad z_{d_i} \sim \text{Cat}(\boldsymbol{\theta}_d)$$
$$\boldsymbol{\phi}_z \sim \text{Dir}(\beta) \qquad w_{d_i} \sim \text{Cat}(\boldsymbol{\phi}_{z_{d_i}})$$

where Dir and Cat stand for a Dirichlet distribution and a categorical distribution, and z_{d_i} is the topic indicator for the i^{th} word w_{d_i} in document d. Inference for LDA is typically performed by variational inference or Gibbs sampling (Blei et al., 2003; Griffiths and Steyvers, 2004; Teh et al., 2006; Porteous et al., 2008; Yao et al., 2009; Foulds et al., 2013).

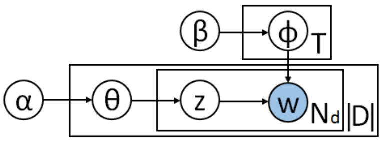


Figure 1: Graphical representation of LDA

When we ignore the Dirichlet priors and apply the Expectation Maximization (EM) algorithm to optimize the likelihood over the document-to-topic and topic-to-word parameters $\theta_{d,z}$ and $\phi_{z,w}$, we obtain the Probabilistic Latent Semantic Analysis model (Hofmann, 1999; Girolami and Kabán, 2003). Optimizing the MAP estimation for the LDA model has been suggested before. Chien and Wu (2008), Asuncion et al. (2009) and Taddy (2012) proposed EM algorithms for estimating $\theta_{d,z}$ and $\phi_{z,w}$, while we use direct gradient-based optimization methods. Sontag and Roy (2011) optimized the MAP estimates of $\phi_{z,w}$ and $\theta_{d,z}$ in turn by integrating out $\theta_{d,z}$ and $\phi_{z,w}$ respectively. We, on the other hand, estimate all parameters jointly in a single optimization step.

In addition to Taddy (2012)'s approach, applying MAP estimation to learn log-linear models for topic models is also found in Eisenstein et al. (2011) and Paul and Dredze (2015). Our MAP model is also defined in log-linear representation. However, unlike our MAP approach, those approaches do not use latent feature word vectors to characterize the topic-to-word distributions.

Furthermore, Berg-Kirkpatrick et al. (2010) proposed a direct optimization approach of the objective function for Hidden Markov Model-like generative models. However, they applied the approach to various unsupervised NLP tasks, such as part-of-speech induction, grammar induction, word alignment, and word segmentation, but not to topic models.

3 Direct MAP estimation approach

In this section, we describe our new direct MAP estimation approach using word vectors for LDA.

Following the likelihood principle, the document-to-topic and topic-to-word distributions $\boldsymbol{\theta}_d$ and $\boldsymbol{\phi}_z$ are determined by maximizing the log likelihood function:

$$\mathcal{L} = \sum_{d \in D} \sum_{w \in W} n_{d,w} \log \sum_z \theta_{d,z} \phi_{z,w} \tag{1}$$

where $n_{d,w}$ is the number of times the word type w appears in document d.

Estimating the parameters $\theta_{d,z}$ and $\phi_{z,w}$ in the original simplex space requires constraints: $\theta_{d,z} \geq 0, \phi_{z,w} \geq 0, \sum_z \theta_{d,z} = 1$ and $\sum_w \phi_{z,w} = 1$. In order to avoid those constraints and to improve estimation efficiency, we transfer the parameters into the natural exponential family parameterization. So we define $\theta_{d,z}$ and $\phi_{z,w}$ as follows:

$$\begin{aligned} \theta_{d,z} &= \frac{\exp(\xi_{d,z})}{\sum\limits_{z'} \exp(\xi_{d,z'})} \\ \phi_{z,w} &= \frac{\exp(\boldsymbol{v}_w \boldsymbol{\cdot} \boldsymbol{\mu}_z + \psi_{z,w})}{\sum\limits_{w' \in W} \exp(\boldsymbol{v}_{w'} \boldsymbol{\cdot} \boldsymbol{\mu}_z + \psi_{z,w'})} \end{aligned} \tag{2}$$

where $\boldsymbol{v}_w$ is the m-dimensional vector associated with word w, while $\boldsymbol{\mu}_z$ is the m-dimensional topic vector associated with topic z. Here $\boldsymbol{v}$ is fixed, and we will learn $\boldsymbol{\mu}$ together with $\boldsymbol{\xi}$ and $\boldsymbol{\psi}$.

With L_2 and L_1 regularizers, we have a new objective function as follows:

$$\begin{aligned} \mathcal{L} = &\sum_{d \in D} \sum_{w \in W} n_{d,w} \log \sum_z \left(\begin{array}{l} \frac{\exp(\xi_{d,z})}{\sum_{z'} \exp(\xi_{d,z'})} \times \\ \frac{\exp(\boldsymbol{v}_w \boldsymbol{\cdot} \boldsymbol{\mu}_z + \psi_{z,w})}{\sum_{w' \in W} \exp(\boldsymbol{v}_{w'} \boldsymbol{\cdot} \boldsymbol{\mu}_z + \psi_{z,w'})} \end{array} \right) \\ &- \sum_{d \in D} \left(\lambda_2 \|\boldsymbol{\xi}_d\|_2^2 + \lambda_1 \|\boldsymbol{\xi}_d\|_1 \right) \\ &- \sum_z \left(\pi_2 \|\boldsymbol{\mu}_z\|_2^2 + \pi_1 \|\boldsymbol{\mu}_z\|_1 \right) \\ &- \sum_z \left(\epsilon_2 \|\boldsymbol{\psi}_z\|_2^2 + \epsilon_1 \|\boldsymbol{\psi}_z\|_1 \right) \end{aligned} \tag{3}$$

The MAP estimate of the model parameters is obtained by maximizing the regularized log likelihood $\mathcal{L}$. The derivatives with respect to the parameters $\xi_{d,z}$ and $\psi_{z,w}$ are:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \xi_{d,z}} = &\sum_{w \in W} n_{d,w} \text{P}(z \mid w, d) - n_d \theta_{d,z} \\ &- 2\lambda_2 \xi_{d,z} - \lambda_1 \text{sign}(\xi_{d,z}) \end{aligned} \tag{4}$$

where $\mathrm{P}(z \mid w, d) = \frac{\theta_{d,z}\phi_{z,w}}{\sum\limits_{z'} \theta_{d,z'}\phi_{z',w}}$, and n_d is the total number of word tokens in the document d.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \psi_{z,w}} &= \sum_{d \in D} n_{d,w} \mathrm{P}(z \mid w, d) \\ &- \phi_{z,w} \sum_{d \in D} \sum_{w' \in W} n_{d,w'} \mathrm{P}(z \mid w', d) \\ &- 2\epsilon_2 \psi_{z,w} - \epsilon_1 \mathrm{sign}(\psi_{z,w})\end{aligned} \tag{5}$$

And the derivative with respect to the j^{th} element of the vector for each topic z is:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu_{z,j}} =& \sum_{d \in D} \sum_{w \in W} n_{d,w} \mathrm{P}(z \mid w, d) \Big(v_{w,j} - \sum_{w' \in W} v_{w',j} \phi_{z,w'} \Big) \\ &- 2\pi_2 \mu_{z,j} - \pi_1 \mathrm{sign}(\mu_{z,j})\end{aligned} \tag{6}$$

We used OWL-QN[1] (Andrew and Gao, 2007) to find the topic vector $\boldsymbol{\mu}_z$ and the parameters $\xi_{d,z}$ and $\psi_{z,w}$ that maximize $\mathcal{L}$.

4 Experiments

To investigate the performance of our new approach, we compared it with two baselines on topic coherence: 1) variational inference LDA (Blei et al., 2003); and 2) Gibbs sampling LDA (Griffiths and Steyvers, 2004). The topic coherence evaluation measures the coherence of the topic-to-word associations, i.e. it directly evaluates how the high-probability words in each topic are semantically coherent (Chang et al., 2009; Newman et al., 2010; Mimno et al., 2011; Stevens et al., 2012; Lau et al., 2014; Röder et al., 2015).

4.1 Experimental setup

We conducted experiments on the standard benchmark 20-Newsgroups dataset.[2]

In addition to converting into lowercase and removing non-alphabetic characters, we removed stop-words found in the stop-word list in the Mallet toolkit (McCallum, 2002). We then removed words shorter than 3 characters or words appearing less than 10 times. Table 1 presents details of the experimental dataset.

Dataset	**#docs**	**#w/d**	**|W|**
20-Newsgroups	18,820	105	20,940

Table 1: Details of the experimental dataset. #docs: number of documents; #w/d: the average number of words per document; |W|: the number of word types.

As pointed out in Levy and Goldberg (2014) and Pennington et al. (2014), the prediction-based methods and count-based methods for learning word vectors are not qualitatively different on a range of semantic evaluation tasks. Thus, we simply use the Word2Vec toolkit[3] (Mikolov et al., 2013) to learn 25-dimensional word vectors on the experimental dataset, using a local 10-word window context.[4]

The numbers of topics is set to 20. For variational inference LDA, we use Blei's implementation.[5] For Gibbs sampling LDA, we use the `jLDADMM` package[6] (Nguyen, 2015) with common hyper-parameters $\beta = 0.01$ and $\alpha = 0.1$ (Newman et al., 2009; Hu et al., 2011; Xie and Xing, 2013). We ran Gibbs sampling LDA for 2000 iterations and evaluated the topics assigned to words in the last sample. We then used the document-to-topic and topic-to-word distributions from the last sample of Gibbs sampling LDA to initialize the parameters $\xi_{d,z}$ and $\psi_{z,w}$ while topic vectors $\boldsymbol{\mu}_z$ are initialized as zero vectors in our MAP learner. For our MAP approach, we set[7] $\lambda_2 = \pi_2 = 0.01$, $\lambda_1 = \pi_1 = 1.0e{-}6$, $\epsilon_2 = 0.1$ and $\epsilon_1 = 0.01$. We report the mean and standard deviation of the results of ten repetitions of each experiment.

4.2 Quantitative analysis

For a quantitative analysis on topic coherence, we use the normalized pointwise mutual information (NPMI) score. Lau et al. (2014) showed that human scores on a word intrusion task are strongly correlated with NPMI. A higher NPMI score indicates that the topic distributions are semantically more coherent.

Given a topic t represented by its top-N topic words $w_1, w_2, ..., w_N$, the NPMI score for t is: $\text{NPMI}(t) = \sum_{1 \leqslant i < j \leqslant N} \frac{\log \frac{\mathrm{P}(w_i, w_j)}{\mathrm{P}(w_i)\mathrm{P}(w_j)}}{-\log \mathrm{P}(w_i, w_j)}$, where the

[1] We employed the OWL-QN implementation from the Mallet toolkit (McCallum, 2002).

[2] We used the "all-terms" version of the 20-Newsgroups dataset available at http://web.ist.utl.pt/acardoso/datasets/ (Cardoso-Cachopo, 2007).

[3] https://code.google.com/p/word2vec/

[4] The parameters of Word2Vec are set to "-cbow 0 -size 25 -window 10 -negative 0 -hs 1."

[5] http://www.cs.princeton.edu/~blei/lda-c/. We used initial value $\alpha = 0.1$ and settings of "var max iter 20, var convergence 1.0e−12, em convergence 1.0e − 8, em max iter 500, alpha estimate".

[6] http://jldadmm.sourceforge.net/

[7] We simply fixed the values of λ_2, π_2, λ_1, π_1, and then varied the values of ϵ_2 and ϵ_1 in $\{0.01, 0.05, 0.1\}$.

Topic 1		Topic 2		Topic 12		Topic 18		Topic 19	
G-LDA	MAP+V	G-LDA	MAP+V	G-LDA	MAP+V	G-LDA	MAP+V	G-LDA	MAP+V
car	car	power	sale	game	game	space	space	medical	medical
writes	cars	sale	power	team	team	nasa	nasa	disease	disease
article	engine	work	**shipping**	year	games	gov	earth	article	health
cars	oil	battery	**offer**	games	year	earth	gov	health	food
engine	speed	radio	battery	hockey	play	writes	launch	drug	drug
good	miles	good	radio	writes	hockey	article	moon	food	cancer
oil	price	high	ground	play	players	launch	orbit	cancer	**doctor**
price	**dealer**	sound	sound	players	season	moon	shuttle	msg	drugs
speed	**ford**	ground	high	season	**win**	orbit	**mission**	drugs	msg
miles	**drive**	writes	**cable**	article	**baseball**	shuttle	**henry**	writes	**patients**

Table 3: Examples of the 10 most probable topical words on the 20-Newsgroups dataset. G-LDA → Gibbs sampling LDA; MAP+V → Our MAP approach using internal word vectors. The words found by G-LDA and not by MAP+V are underlined. The words found by MAP+V but not by G-LDA are in **bold**.

Method	**Top-10**	**Top-15**	**Top-20**
V-LDA	-4.2 ± 0.4	-12.2 ± 0.6	-24.1 ± 0.6
G-LDA	-4.2 ± 0.4	-11.7 ± 0.7	-22.9 ± 0.9
MAP-O	-3.8 ± 0.5	-10.8 ± 0.6	-22.1 ± 1.2
MAP+V	**-3.4 ± 0.3**	**-10.1 ± 0.7**	**-20.6 ± 1.0**
Improve.	0.8	1.6	2.3

Table 2: NPMI scores (mean and standard deviation) on the 20-Newsgroups dataset with different numbers of top topical words; V-LDA → Variational inference LDA; G-LDA → Gibbs sampling LDA; MAP-O → Our MAP learner where we fix topic vectors $\boldsymbol{\mu}$ as zero vectors and only learn parameters $\boldsymbol{\xi}$ and $\boldsymbol{\psi}$; MAP+V → Our MAP learner where we learn $\boldsymbol{\mu}$ together with $\boldsymbol{\xi}$ and $\boldsymbol{\psi}$. The *Improve.* row denotes the absolute improvement accounted for MAP+V over the best result produced by the baselines V-LDA and G-LDA.

probabilities are derived from a 10-word sliding window over an external corpus.[8] The NPMI score for a topic model is the average score for all topics.

Table 2 shows that our approach using internal word vectors MAP+V produces significantly higher[9] NPMI scores than the baseline variational inference LDA and Gibbs sampling LDA models. So this indicates that the word vectors containing internal context information from experimental dataset can help to improve topic coherence.

4.3 Qualitative analysis

This section provides an example of how our approach improves topic coherence. Table 3 compares the top-10 words produced by the baseline Gibbs sampling LDA and our MAP+V approach on the 20-Newsgroups dataset. It is clear that all top-10 words learned with our MAP+V model are qualitatively more coherent. For example, topic 19 of the Gibbs sampling LDA model consists of words related to "medicine" together with other unrelated words, whereas our MAP+V approach produced a purer topic 19 only about "medicine."

On 20-Newsgroups dataset, it is common that the baseline variational inference LDA and Gibbs sampling LDA models include the frequent words such as ”writes” and ”article” as top topical words in many topics. However, our MAP+V model using the internal word vectors is able to exclude these words out of the top words in these topics.

5 Conclusions and future work

In this paper, we proposed a new approach of fully direct MAP estimation for the LDA topic model inference, incorporating latent feature representations of words. Preliminary results show that the latent feature representations trained from the experimental topic-modeling corpus can improve the topic-to-word mapping.

In future work, we plan to investigate the effects of the context window size as well as the size of the word vectors further. In addition, we plan to test our approach on a range of different datasets. We also plan to compare the presented results with Nguyen et al. (2015)'s model using internal word vectors. Even though we learn the internal word vectors from the experimental dataset, we believe that it is worth trying to initialize them from vectors learned from an external corpus, thus also incorporating generalizations from that corpus.

[8] We use the English Wikipedia dump of July 8, 2014, containing 4.6 million articles as our external corpus.

[9] Using the two sample Wilcoxon test, the improvement is significant ($p < 0.01$).

Acknowledgments

The first author is supported by an International Postgraduate Research Scholarship and a NICTA NRPA Top-Up Scholarship.

References

Galen Andrew and Jianfeng Gao. 2007. Scalable Training of L1-regularized Log-linear Models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. 2009. On Smoothing and Inference for Topic Models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless Unsupervised Learning with Features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590.

David M. Blei and Jon D. McAuliffe. 2008. Supervised Topic Models. In *Advances in Neural Information Processing Systems 20*, pages 121–128. MIT Press.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

David M. Blei. 2012. Probabilistic Topic Models. *Communications of the ACM*, 55(4):77–84.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A Novel Neural Topic Model and Its Supervised Extension. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2210–2216.

Ana Cardoso-Cachopo. 2007. Improving Methods for Single-label Text Categorization. PhD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22*, pages 288–296.

Jen-Tzung Chien and Meng-Sung Wu. 2008. Adaptive Bayesian Latent Semantic Analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):198–207.

Jacob Eisenstein, Amr Ahmed, and Eric Xing. 2011. Sparse Additive Generative Models of Text. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1041–1048.

James Foulds, Levi Boyles, Christopher DuBois, Padhraic Smyth, and Max Welling. 2013. Stochastic Collapsed Variational Bayesian Inference for Latent Dirichlet Allocation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 446–454.

Mark Girolami and Ata Kabán. 2003. On an Equivalence Between PLSI and LDA. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 433–434.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Yuening Hu, Jordan Boyd-Graber, and Brianna Satinoff. 2011. Interactive Topic Modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 248–257.

Mark Johnson. 2010. PCFGs, Topic Models, Adaptor Grammars and Learning Topical Collocations and the Structure of Proper Names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1157.

Han Jey Lau, David Newman, and Timothy Baldwin. 2014. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems 27*, pages 2177–2185.

Andrew McCallum. 2002. MALLET: A Machine Learning for Language Toolkit.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing Semantic Coherence in Topic Models.

In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2009. Distributed Algorithms for Topic Models. *The Journal of Machine Learning Research*, 10:1801–1828.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving Topic Models with Latent Feature Word Representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Dat Quoc Nguyen. 2015. jLDADMM: A Java package for the LDA and DMM topic models. http://jldadmm.sourceforge.net/.

Michael Paul and Mark Dredze. 2015. SPRITE: Generalizing Topic Models with Structured Priors. *Transactions of the Association for Computational Linguistics*, 3:43–57.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 399–408.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Replicated Softmax: an Undirected Topic Model. In *Advances in Neural Information Processing Systems 22*, pages 1607–1614.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013. Parsing with Compositional Vector Grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465.

David Sontag and Dan Roy. 2011. Complexity of Inference in Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 24*, pages 1008–1016.

Nitish Srivastava, Ruslan Salakhutdinov, and Geoffrey Hinton. 2013. Modeling Documents with a Deep Boltzmann Machine. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 616–624.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring Topic Coherence over Many Models and Many Topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961.

Matthew A. Taddy. 2012. On Estimation and Selection for Topic Models. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*.

Yee W Teh, David Newman, and Max Welling. 2006. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 19*, pages 1353–1360.

Hanna M Wallach. 2006. Topic Modeling: Beyond Bag-of-Words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984.

Xuerui Wang, Andrew McCallum, and Xing Wei. 2007. Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 697–702.

Pengtao Xie and Eric P. Xing. 2013. Integrating Document Clustering and Topic Modeling. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 694–703.

Pengtao Xie, Diyi Yang, and Eric Xing. 2015. Incorporating Word Correlation Knowledge into Topic Modeling. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 725–734.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A Biterm Topic Model for Short Texts. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 1445–1456.

Yi Yang, Doug Downey, and Jordan Boyd-Graber. 2015. Efficient Methods for Incorporating Knowledge into Topic Models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 308–317.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient Methods for Topic Model Inference on Streaming Document Collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 937–946.

Understanding Engagement with Insurgents through Retweet Rhetoric

Joel Nothman[1] **Atif Ahmad**[1] **Christoph Breidbach**[1]
David Malet[2] **Timothy Baldwin**[1]
[1] Department of Computing and Information Systems
[2] School of Social and Political Sciences
The University of Melbourne
{jnothman,atif,christoph.breidbach,david.malet,tbaldwin}
@unimelb.edu.au

Abstract

Organisations — including insurgent movements — harness social media to engage potential consumers. They evoke sympathetic (and antipathic) response; content sharing engenders affinity and community. We report on a pilot study of presumed rhetorical intent for statuses retweeted by a set of suspected Islamic State-sympathetic Twitter accounts. This annotation is orthogonal to prior opinion mining work focused on sentiment or stance expressed in a debate, and suggests a parallel to dialogue act classification applied to retweeting. By exploring the distribution of rhetoric among Islamic State-sympathetic and general users, we also hope to identify trends in IS social media use and user roles.

1 Introduction

Social media has become an important platform for organisations and communities seeking to engage with adherents and the wider public. Through it we may follow individuals as their ideas and affiliations change, expressed through conversation, broadcast, and rebroadcast. Social scientists are keen to understand how individuals are transformed in this process of engagement: how this is effected by the organisation, and how it is realised in individual behaviour.

Recent media and scholarly studies have highlighted the use of social media by insurgent organisations. Understanding and tracking these activities is of particular interest to law enforcement and policy makers, as well as political scientists studying the nature of conflict, in terms of both monitoring and comprehending insurgent activities.

This work presents a new annotation model of partisan retweets (RTs), as "rhetorical acts". It pilots a study of content rebroadcast by suspected IS-sympathetic Twitter users. We develop an annotation schema to capture the attitude of a partisan user when retweeting content, and are able to analyse trends with respect to popularity, and transmission into/out of the IS network.

For our pilot set of suspected IS-sympathetic accounts, we find that 58% of RTs are evocative; these divide almost equally between expressing pride in the movement, expressing indignation at oppression, and transmitting religious and partisan mythology. Most others (22%) share general content, while 3% manage the ISIS Twitter network under suspension.

2 Background

Insurgent movements exploit the decentralised and colloquial nature of social media to counter mainstream narratives (Thompson, 2011; Bernatis, 2014). Berger and Morgan's (2015) seminal study of IS Twitter accounts describes their network structure and measures status sharing from within or outside the network, but gives little attention to content. Klausen (2015) analyses 10 statuses for each of 59 IS accounts, finding 40% of them deal with religious instruction, and a further 40% report from battle. Our focus on RT intent highlights the dissemination of rhetoric and its affect, classified at finer granularity than this prior work.

We investigate the construction of partisan rhetoric through distributed social media activity, while opinion mining of partisan text has largely followed Lin et al. (2006) in addressing the task of discriminating distinct points of view, in various domains (Somasundaran and Wiebe, 2009; Al Khatib et al., 2012) and granularities (Abu-Jbara et al., 2012). More recent work investigates the language used to frame discussion of contentious topics to appeal to audience values (Card et al., 2015; Baumer et al., 2015; Tsur et al., 2015), building on the premise that sentiment can be detected in statements that are not subjective or eval-

Joel Nothman, Atif Ahmad, Christoph Breidbach, David Malet and Timothy Baldwin. 2015. Understanding engagement with insurgents through retweet rhetoric . In *Proceedings of Australasian Language Technology Association Workshop*, pages 122−127.

uative (Greene and Resnik, 2009). We similarly model partisan rhetorical processes that aim to engage a sympathetic audience.

Communicative units may be analysed as *dialogue acts*, which classify the intended effect on an addressee (e.g. Core and Allen, 1997). This has been applied to Twitter conversation threads with coarse classes — STATUS, REACTION, QUESTION, etc. — and with a fine-grained act hierarchy (Zarisheva and Scheffler, 2015); Zhang et al. (2011) broadly classify isolated tweets. We depart from that work to analyse rebroadcasting, not authoring, through a partisan lens.

3 Retweets as rhetoric

Propagating broadcast content has become a key feature of social media, and we choose it as a lens for analysing the IS Twitter network. Initial attempts at analysing a sample of tweets by IS-affiliated users suggest it is too noisy: the majority of statuses are poor in rhetorical and evocative content, and tend to be hard to interpret without context. In contrast, the act of propagating a status — *retweeting* in Twitter — inherently declares that it is of interest beyond its author, and usually implies that a message is encapsulated within the shared status, such that little discourse context is required to understand it. Sharing a status is a rhetorical act, although the attitude of the retweeter — our focus — often differs from that of the author.

4 Annotation schema

We examine a sample of RTs by suspected IS supporters, asking: what was the user expressing by rebroadcasting this status *assuming they are sympathetic towards IS*? We develop a shallow hierarchical schema for high coverage but reasonable robustness. At its root we distinguish between: EVOCATIVE/INSTRUCTIVE (along the lines of traditional "propaganda"); OPERATION FACILITATION; GENERAL CONTENT; and SPAM.

In some cases there is NOT ENOUGH INFORMATION to determine the category of a status. This occurs where conversational context is necessary; or where an image attached to the status was necessary, but is no longer available, frequently due to suspension of its poster.

4.1 EVOCATIVE/INSTRUCTIVE

We assume much of the content is evocative to the retweeter, as with other social media sharing (Berger and Milkman, 2012), even when it is objectively stated by the original author. We identify the following subcategories.

PRIDE: usually good news for IS, often evoking pride in IS government or land (1), military might (2)–(3), or victory (4):

(1) The building of #IS new college of medicine in ar-Raqqah #Syria *[image]*

(2) Qamishli: 4 members of the pro-Assad Maghaweer militia have defected and have now joined the Islamic State.

(3) From a small group of Jihadists surrounded in #Fallujah in 2004 into a large Islamic State that controls large parts of #Syria #Iraq in 2014

(4) BREAKING: #IslamicState shot down a warplane above kuwairs military airport !!!!! Al-Hamdulillah

INDIGNATION: expressed directly (5)–(6), or implied from news of loss (7):

(5) For all of those who normalize assad's mass killings deserve to be flayed and disemboweled alive

(6) shia rafidis slaves of kuffar, harassing sunni Muslims in Baghdad *[image]*

(7) Bismillah. iPICTURE - The U.S. organised Shi'a death squads stormed Ibn Tamiyyah mosque in Baghdad & kidnaps Sunni's *[image]*

DERISION: develops an us-vs-them dichotomy by mocking various enemies:

(8) America's track record: Successfully wiped out the Native Americans, enslaved the entire African continent, and now fighting Islam/Muslims.

(9) This is why Peshmerga cant win they need to fly all the way to Germany for treatment (This is not a joke) #IS *[image]*

INSTRUCTION: distribution of ideological materials, often religious (10), or claiming authenticity (11)–(13).

(10) The lack of showing Bara' (disavowal) towards the polytheists and apostates: Dr al-Jazouli *[url:youtube.com]*

(11) If u're a scholar in Saudi Arabia and not in jail. U're NT truthful then.

(12) How can the IS be anti-kurdish,when a large part of the attacking troops in Kobane are Kurds themselves?Don't believe the media-propaganda!

(13) #IS written by S.Qutub,taught by A.Azzam, globalized by Ben Laden, being real by Zarkawi, carried out by the 2 Baghdadi:Abu Omar & Abu Baker

4.2 Operational

Intended to facilitate online operations, particularly maintaining Twitter network (and web sites) under adversity (14), including announcing new accounts following suspension (15):

(14) Attention bro n sisters please be careful of following people on Twitter, Trolls are changing strategy: make a pro IS account 2 lure muslims

(15) Follow ReTweet Spread n Support @AbuAdamIsBack @AbuAdamIsBack @AbuAdamIsBack

4.3 General

Creates "rapport" with followers through culture (16), humour (17), conversation (18), political news (19); or external media reports about IS and affiliates, without clear evocative aspect (20):

(16) I love how the remix isn't just thrown together. They actually put effort into making the verses go together

(17) Hahaha "pissing" myself laughing.. India launches cow urine based soda. *[url:eshowbizbuzz]*

(18) Happy Bday Sami!! @SamiAlJaber *[image]*

(19) UK spies have scanned the internet connections of entire countries: *[url:engadget.com]*

(20) Very isnightful interview with British-Pakistani Jihadi in #TTP by @Quickieleaks Must read: *[url:londonprogressivejournal.com]*

4.4 Spam

Spam content, unrelated to IS:

(21) #workoutwednesday Back & Abs what you doing today? "Girls just wanna tank" @Bodybuildingcom *[image][url:fit-kat.com]*

(22) #GAIN_FOLLOWERS #MENTIONS #FOLLOWME & @Gamma_Monkey @jockomo141 @PATOO_S @Trans1110 @Sammi_Gemini @MREESE06 @Retweetsjp & ALL #RTS #TFB

5 Data

Affiliate accounts Prior work has painstakingly identified IS-affiliated accounts (Berger and Morgan, 2015), or has shown the success of simple heuristics (Magdy et al., 2015). The latter finds that Twitter accounts using unabbreviated forms of the IS name in Arabic-language tweets are very frequently IS supporters. This heuristic does not apply trivially to English-language tweets.

We instead combine noisy lists of suspected accounts: *LuckyTroll.club* was collected by counter-IS hacktivists on Twitter and published online,[1] which we scraped from 2015-03-16 until 2015-05-18, yielding 36,687 accounts. Another anonymous list of 555 accounts labelled *#GoatsAgainstIsis* was published on `ghostbin.com` and linked from a hacktivist Twitter account. We add 36 usernames from two English-language purported IS *guide books* available from the Internet Archive (Anon., 2015a,b). Despite observing false entries — members of rival groups and unlikely Jihadis — we make no attempt to clean them.

Twitter stream Investigating IS on Twitter presents a number of challenges, particularly since Twitter began suspending affiliated accounts from mid-2014. Once suspended, Twitter's API provides no information about an account, so traditional social media analysis with follower graphs or extensive activity histories are not available. Prior work has retrieved IS user histories before their suspension, but this data is not available to us; still, we seek to make the scope of the project as broad as possible, in including both suspended and active accounts.

We use tweets collected from the Twitter Streaming API from 2014-01-01 to 2015-03-20,[2] analysed regardless of eventual suspension/retraction. An annotated status must satisfy the following criteria: (1) posted by a user in our set of suspected affiliate accounts; (2) produced using the official Twitter RT mechanism; and (3) recognised by Twitter as being in English.

We remove any duplicate RTs[3] and reduce skew to major content producers by sampling in proportion to the square root of the number of tweets by each originating author. A single annotator labelled 400 statuses with RT intent.

6 Experiments and results

Annotator agreement A second annotator, an expert in jihadist ideology, coded 100 tweets after a brief introduction to the schema. On coarse categories, the annotators agree reasonably often, $\kappa = 0.40$. This second annotator overgenerated spam labels, including various off-topic posts, e.g. news about North American weather events; conflating general and spam labels yields $\kappa = 0.45$. At the finest granularity (e.g. "religious instruction", "general humour"), agreement is a weaker $\kappa = 0.28$. Disagreement often results from content

[1] `https://luckytroll.club/daesh`

[2] Disruptions leave much of March–May 2015 and February 2015 absent.

[3] For annotation; for analysis, repetition is informative.

	#	sus author	> 2	> 2 sus
EVOCATIVE/INSTRUCTIVE	232	56	96	16
– PRIDE	64	22	25	3
– INDIGNATION	65	10	32	6
– DERISION	15	5	6	1
– INSTRUCTION	66	12	27	4
– OTHER	22	7	6	2
OPERATIONAL	14	8	4	0
GENERAL	89	5	45	6
– ABOUT IS/AFFILIATES	7	1	2	0
– POLITICS/WARFARE	35	2	12	1
– HUMOUR	5	2	5	2
– CONVERSATION	10	0	2	0
– OTHER	32	0	24	3
SPAM	15	4	7	1
NOT ENOUGH INFO	49	17	21	10
TOTAL	400	90	173	33

Table 1: Distribution of RT intent: overall; for statuses with suspected authors; for statuses with over two sampled RTs by any/suspected users.

about groups towards which IS followers are sympathetic; one annotator saw indignation in descriptions of Gazan suffering, the second saw a general informative intent. Further schema refinement and training will hopefully reduce disagreement.

Intent distribution We analyse the RT intent distribution with respect to popularity and whether the author or retweeters are IS suspects. We regard as popular any RTs that are thrice sampled in the stream.[4] Here, suspects include those listed above, plus any accounts deactivated by 2015-09-30, often due to suspension.

Granular annotation frequencies are shown in Table 1. RTs by IS users are dominated by messages that they would find evocative or instructive (58%). Most are divided equally between pride (mostly about military strength), indignation, and instruction in group mythology. Indignation is characterised by being widely spread, beyond IS suspects, and often originating outside that network. This accords with studies showing that insurgents see themselves as addressing communal grievances (Hafez, 2007; Mironova et al., 2014). GENERAL content, often political and sourced from non-suspects, is also frequently retweeted, while a small portion (3%) of RTs maintain IS Twitter operations. Overall these distributions hint that IS RTs use religious-cultural affect and political interest as a guide towards insurgent engagement.

7 Discussion

Inter-annotator agreement shows that likely intent behind an IS affiliate's RT is often determinate. Reviewing users' own remarks on their RTs might provide more robust evaluation of our annotations.[5] Suspensions make this difficult, suggesting that this task be attempted with less-controversial affiliations. We are further hampered by suspect lists collected by an unknown process that may consider the rhetoric of the user, perhaps biasing our results, e.g. derisive RTs are frequently authored and distributed by suspects.

RTs about affiliated and rival groups are among the most ambiguous for our task. Damage to a rival jihadist organisation in (23) may be a source of both indignation and pride (or schadenfreude); (24)'s apologetics for terror in the west is not clearly apologetics for IS; and though (25) literally expresses solidarity, it may pity its subject bereft of Islamic sovereignty. Such cases highlight that intent is affected by the relationship between author, retweeter and theme, suggesting future analysis akin to Verbal Response Modes (Stiles, 1992).

(23) #JabhatNusra A headquarter of #JabhatAnNusra which was bombed by the Crusaders in #Kafrdarian #Idlib (1) *[image]*

(24) #CharlieHebdo Operation wasnt a gun rampage. Gunmen had a list with targets to be assassinated rest of the staff & civilians were free to go

(25) Oh Allah bless our brothers in the UK who hav held the rope of haq.. Even in difficult tyms and never compromised.. Ya Allah bless them...

8 Conclusion

We have presented an initial treatment of the act of social media rebroadcasting as akin to a speech act, laden with rhetorical intent in the context of partisan propaganda. We hope our work lights the way towards a more general model of this quintessential social media communicative act. Though we leave automatic classification to future work, large scale analysis of IS RT intent may allow us to analyse different types of IS-affiliated users, and identify changes in rhetoric over time and place that are indicative of radicalisation.

Acknowledgements

This work was supported by the Melbourne School of Government.

[4] The unknown, variable sampling rate — historically 10% of all tweets — makes this a weak heuristic.

[5] Since mid-2014, Twitter allows users add remarks when retweeting. The original status is not provided on the stream in such cases, and so is inaccessible after suspension.

References

Amjad Abu-Jbara, Pradeep Dasigi, Mona Diab, and Dragomir Radev. 2012. Subgroup detection in ideological discussions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–409.

Khalid Al Khatib, Hinrich Schütze, and Cathleen Kantner. 2012. Automatic detection of point of view differences in Wikipedia. In *Proceedings of COLING 2012*, pages 33–50.

Anon. 2015a. *Hijrah to the Islamic State*. The Internet Archive. `https://archive.org/details/GuideBookHijrah2015-ToTheIslamicState`.

Anon. 2015b. *The Islamic State*. The Internet Archive. `https://archive.org/details/TheIslamicState2015-FullEbook`.

Eric Baumer, Elisha Elovic, Ying Qin, Francesca Polletta, and Geri Gay. 2015. Testing and comparing computational approaches for identifying the language of framing in political news. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1472–1482.

J.M. Berger and Jonathon Morgan. 2015. *The ISIS Twitter census: defining and describing the population of ISIS supporters on Twitter*. Number 20 in The Brookings Project on U.S. Relations with the Islamic World. Brookings Institution.

Jonah Berger and Katherine L. Milkman. 2012. What makes online content viral? *Journal of Marketing Research*, 49(2):192–205.

Vincent Bernatis. 2014. The Taliban and Twitter: Tactical reporting and strategic messaging. *Perspectives on Terrorism*, 8(6).

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. The media frames corpus: Annotations of frames across issues. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 438–444.

Mark Core and James Allen. 1997. Coding dialogs with the DAMSL annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35.

Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511.

Mohammed Hafez. 2007. *Suicide Bombers in Iraq: The Strategy and Ideology of Martyrdom*. United States Institute of Peace Press.

Jytte Klausen. 2015. Tweeting the Jihad: Social media networks of western foreign fighters in syria and iraq. *Studies in Conflict & Terrorism*, 38(1):1–22.

Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. 2006. Which side are you on? Identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 109–116.

Walid Magdy, Kareem Darwish, and Ingmar Weber. 2015. #FailedRevolutions: Using Twitter to study the antecedents of ISIS support. *arXiv CoRR*, abs/1503.02401.

Vera Mironova, Loubna Mrie, and Sam Whitt. 2014. The motivations of Syrian Islamist fighters. *CTC Sentinel*, 7(10):15–17.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 226–234.

William B Stiles. 1992. *Describing talk: a taxonomy of verbal response modes*. SAGE Series in Interpersonal Communication. SAGE Publications.

Robin L. Thompson. 2011. Radicalization and the use of social media. *Journal of Strategic Security*, 4(4):167–190.

Oren Tsur, Dan Calacci, and David Lazer. 2015. A frame of mind: Using statistical models for detection of framing and agenda setting campaigns. 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), pages 1629–1638.

Elina Zarisheva and Tatjana Scheffler. 2015. Dialog act annotation for Twitter conversations. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 114–123.

Renxian Zhang, Dehong Gao, and Wenjie Li. 2011. What are tweeters doing: Recognizing speech acts in Twitter. In *Analyzing Microtext: Papers from the 2011 AAAI Workshop*.

A comparison and analysis of models for event trigger detection

Shang Chun Sam Wei
School of Information Technologies
University of Sydney
NSW 2006, Australia
swei4829@uni.sydney.edu.au

Ben Hachey
School of Information Technologies
University of Sydney
NSW 2006, Australia
ben.hachey@gmail.com

Abstract

Interpreting event mentions in text is central to many tasks from scientific research to intelligence gathering. We present an event trigger detection system and explore baseline configurations. Specifically, we test whether it is better to use a single multi-class classifier or separate binary classifiers for each label. The results suggest that binary SVM classifiers outperform multi-class maximum entropy by 6.4 points F-score. Brown cluster and WordNet features are complementary with more improvement from WordNet features.

1 Introduction

Events are frequently discussed in text, e.g., criminal activities such as violent attacks reported in police reports, corporate activities such as mergers reported in business news, biological processes such as protein interactions reported in scientific research. Interpreting these mentions is central to tasks like intelligence gathering and scientific research. Event extraction automatically identifies the triggers and arguments that constitute a textual mention of an event in the world. Consider:

Bob **bought** the book from Alice.

Here, a trigger – "bought" (*Transaction.Transfer–Ownership*) – predicates an interaction between the arguments – "Bob" (*Recipient*), "the book" (*Thing*) and "Alice" (*Giver*). We focus on the trigger detection task, which is the first step in event detection and integration.

Many event extraction systems use a pipelined approach, comprising a binary classifier to detect event triggers followed by a separate multi-class classifier to label the type of event (Ahn, 2006). Our work is different in that we use a single classification step with sub-sampling to handle data skew. Chen and Ji (2009) use Maximum Entropy (ME) classifier in their work. However, their approach is similar to (Ahn, 2006) where they identify the trigger first then classify the trigger at later stage. Kolya et al. (2011) employ a hybrid approach by using Support Vector Machine (SVM) classifier and heuristics for event extraction.

We present an event trigger detection system that formulates the problem as a token-level classification task. Features include lexical and syntactic information from the current token and surrounding context. Features also include additional word class information from Brown clusters, WordNet and Nomlex to help generalise from a fairly small training set. Experiments explore whether multi-class or binary classification is better using SVM and ME.

Contributions include: (1) A comparison of binary and multi-class versions of SVM and ME on the trigger detection task. Experimental results suggest binary SVM outperform other approaches. (2) Analysis showing that Brown cluster, Nomlex and WordNet features contribute nearly 10 points F-score; WordNet+Nomlex features contribute more than Brown cluster features; and improvements from these sources of word class information increase recall substantially, sometimes at the cost of precision.

2 Event Trigger Detection Task

We investigate the event trigger detection task from the 2015 Text Analysis Conference (TAC) shared task (Mitamura and Hovy, 2015). The task defines 9 event types and 38 subtypes such as *Life.Die, Conflict.Attack, Contact.Meet*. An event trigger is the smallest extent of text (usually a word or short phrase) that predicates the occurrence of an event (LDC, 2015).

In the following example, the words in bold trigger *Life.Die* and *Life.Injure* events respectively:

Sam Shang Chun Wei and Ben Hachey. 2015. A comparison and analysis of models for event trigger detection . In *Proceedings of Australasian Language Technology Association Workshop*, pages 128−132.

The explosion **killed** *7 and* **injured** *20.*

Note that an event mention can contain multiple events. Further, an event trigger can have multiple events. Consider:

The **murder** *of John.*

where "murder" is the trigger for both a *Conflict.Attack* event and a *Life.Die* event. Table 1 shows the distribution of the event subtypes in the training and development datasets.

3 Approach

We formulate event trigger detection as a token-level classification task. Features include lexical and semantic information from the current token and surrounding context. Classifiers include binary and multi-class versions of SVM and ME.

As triggers can be a phrase, we experimented with Inside Outside Begin 1 (IOB1) and Inside Outside Begin 2 (IOB2) encodings (Sang and Veenstra, 1999). Table 2 contains an example illustrating the two schemes. Preliminary results showed little impact on accuracy. However, one of the issues with this task is data sparsity. Some event subtypes have few observations in the corpus. IOB2 encoding increases the total number of categories for the dataset. Thus make the data sparsity issue worse. Therefore we use the IOB1 encoding for the rest of the experiments.

Another challenge is that the data is highly unbalanced. Most of the tokens are not event triggers. To address this, we various subsets of negative observations. Randomly sampling 10% of the negative examples for training works well here.

3.1 Features

All models used same rich feature sets. The features are divided into three different groups.

Feature set 1 (FS1): Basic features including following. (1) Current token: Lemma, POS, named entity type, is it a capitalised word. (2) Within the window of size two: unigrams/bigrams of lemma, POS, and name entity type. (3) Dependency: governor/dependent type, governor/dependent type + lemma, governor/dependent type + POS, and governor/dependent type + named entity type.

Feature set 2 (FS2): Brown cluster trained on the Reuters corpus (Brown et al., 1992; Turian et al., 2010) with prefix of length 11, 13 and 16.[1]

Feature set 3 (FS3): (1) WordNet features including hypernyms and synonyms of the current token. (2) Base form of the current token extracted from Nomlex (Macleod et al., 1998).[2]

Event Subtype	Train	Dev
Business.Declare-Bankruptcy	30	3
Business.End-Org	11	2
Business.Merge-Org	28	0
Business.Start-Org	17	1
Conflict.Attack	541	253
Conflict.Demonstrate	162	38
Contact.Broadcast	304	112
Contact.Contact	260	77
Contact.Correspondence	77	18
Contact.Meet	221	23
Justice.Acquit	27	3
Justice.Appeal	25	12
Justice.Arrest-Jail	207	79
Justice.Charge-Indict	149	41
Justice.Convict	173	49
Justice.Execute	51	15
Justice.Extradite	62	1
Justice.Fine	53	2
Justice.Pardon	221	18
Justice.Release-Parole	45	28
Justice.Sentence	118	26
Justice.Sue	54	1
Justice.Trial-Hearing	172	24
Life.Be-Born	13	6
Life.Die	356	157
Life.Divorce	45	0
Life.Injure	63	70
Life.Marry	60	16
Manufacture.Artifact	18	4
Movement.Transport-Artifact	52	18
Movement.Transport-Person	390	125
Personnel.Elect	81	16
Personnel.End-Position	130	79
Personnel.Nominate	30	5
Personnel.Start-Position	60	17
Transaction.Transaction	34	17
Transaction.Transfer-Money	366	185
Transaction.Transfer-Ownership	233	46

Table 1: Event subtype distribution.

[1] `http://metaoptimize.com/projects/wordreprs/`

[2] `http://nlp.cs.nyu.edu/nomlex/`

Word	IOB1	IOB2
He	O	O
has	O	O
been	O	O
found	I-Justice.Convict	B-Justice.Convict
guilty	I-Justice.Convict	I-Justice.Convict
for	O	O
the	O	O
murder	I-Life.Die	B-Life.Die
.	O	O

Table 2: IOB1 and IOB2 encoding comparison. "B" represents the first token of an event trigger. "I" represents a subsequent token of a multi-word trigger. "O" represents no event.

3.2 Classifiers

We train multi-class ME and SVM classifiers to detect and label events. L-BFGS (Liu and Nocedal, 1989) is used as the solver for ME. The SVM uses a linear kernel. We also compare binary versions of ME and SVM by building a single classifier for each event subtype.

4 Experimental setup

4.1 Dataset

The TAC 2015 training dataset (LDC2015E73) is used for the experiment. The corpus has a total of 158 documents from two genres: 81 newswire documents and 77 discussion forum documents. Preprocessing includes tokenisation, sentence splitting, POS tagging, named entity recognition, constituency parsing and dependency parsing using Stanford CoreNLP 3.5.2.[3]

The dataset is split into 80% for training (126 documents) and 20% for development (32 documents. Listed in Appendix A).

4.2 Evaluation metric

Accuracy is measured using the TAC 2015 scorer.[4] Precision, recall and F-score are defined as:

$$P = \frac{TP}{N_S}; R = \frac{TP}{N_G}; F1 = \frac{2PR}{P+R}$$

where TP is the number of correct triggers (true positives), N_S is the total number of predicted system mentions, and N_G is the total number of annotated gold mentions. An event trigger is counted as correct only if the boundary, the event type and the event subtype are all correctly identified. We report micro-averaged results.

5 Results

Table 3 shows the results from each classifier. The binary SVMs outperform all other models with an F-score of 55.7. The score for multi-class SVM is two points lower at 53.2. Multi-class and binary ME comes next with binary performing worst.

System	P	R	F1
Multi-class ME	62.2	40.8	49.2
Multi-class SVM	55.6	50.9	53.2
Binary ME	77.8	28.2	41.4
Binary SVM	64.7	48.9	**55.7**

Table 3: System performance comparison.

5.1 Feature set

We perform a cumulative analysis to quantify the contribution of different feature sets. Table 4 shows that feature set 2 (Brown cluster) helped with recall sometimes at the cost of precision. The recall is further boosted by feature set 3 (WordNet and Nomlex). However, the precision dropped noticeably for SVM models.

System	P	R	F1
Multi-class systems			
ME FS1	54.1	16.9	25.8
ME FS1+FS2	57.8	21.3	31.1
ME FS1+FS2+FS3	62.2	40.8	49.2
SVM FS1	62.1	35.3	45.0
SVM FS1+FS2	60.9	39.3	47.8
SVM FS1+FS2+FS3	55.6	50.9	53.2
Binary systems			
ME FS1	64.7	6.1	11.2
ME FS1+FS2	72.7	10.1	17.8
ME FS1+FS2+FS3	77.8	28.2	41.4
SVM FS1	71.0	34.2	46.2
SVM FS1+FS2	70.5	38.4	49.7
SVM FS1+FS2+FS3	64.7	48.9	**55.7**

Table 4: Feature sets comparison.

5.2 Performance by event subtype

Figure 1 shows how classifiers perform on each event subtype. Binary SVM generally has better recall and slightly lower precision. Hence, the overall performance of the model improves.

[3] `http://nlp.stanford.edu/software/corenlp.shtml`

[4] `http://hunterhector.github.io/EvmEval/`

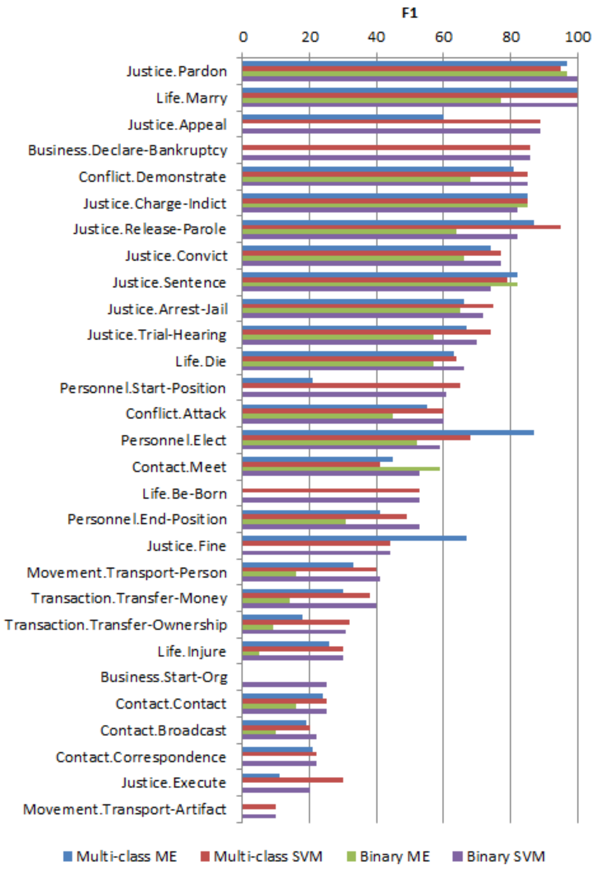


Figure 1: Performance by subtype.

5.3 Error analysis

We sampled 20 precision and twenty recall errors from the binary SVM classifier. 40% of precision errors require better modelling of grammatical relations, e.g., labelling “focus has moved” as a transport event. 35% require better use of POS information, e.g., labelling “said crime” as a contact event. 65% of recall errors are tokens in multiword phrases, e.g., “going to jail”. 45% are triggers that likely weren’t seen in training and require better generalisation strategies. Several precision and recall errors seem to actually be correct.

6 Conclusion

We presented an exploration of TAC event trigger detection and labelling, comparing classifiers and rich features. Results suggest that SVM outperforms maximum entropy and binary SVM gives the best results. Brown cluster information increases recall for all models, but sometimes at the cost of precision. WordNet and Nomlex features provide a bigger boost, improving F-score by 6 points for the best classifier.

Acknowledgements

Sam Wei is funded by the Capital Markets Cooperative Research Centre. Ben Hachey is the recipient of an Australian Research Council Discovery Early Career Researcher Award (DE120102900)

References

David Ahn. 2006. The stages of event extraction. In *COLING-ACL Workshop on Annotating and Reasoning About Time and Events*, pages 1–8.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Zheng Chen and Heng Ji. 2009. Language specific issue and feature exploration in chinese event extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–212.

Anup Kumar Kolya, Asif Ekbal, and Sivaji Bandyopadhyay. 2011. A hybrid approach for event extraction and event actor identification. In *International Conference on Recent Advances in Natural Language Processing*, pages 592–597.

LDC, 2015. *Rich ERE Annotation Guidelines Overview*. Linguistic Data Consortium. Version 4.1. Accessed 14 November 2015 from `http://cairo.lti.cs.cmu.edu/kbp/2015/event/summary_rich_ere_v4.1.pdf`.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. Nomlex: A lexicon of nominalizations. In *Euralex International Congress*, pages 187–193.

Teruko Mitamura and Eduard Hovy, 2015. *TAC KBP Event Detection and Coreference Tasks for English*. Version 1.0. Accessed 14 November 2015 from `http://cairo.lti.cs.cmu.edu/kbp/2015/event/Event_Mention_Detection_and_Coreference-2015-v1.1.pdf`.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semisupervised learning. In *Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Appendix A: Development set document IDs

```
3288ddfcb46d1934ad453afd8a4e292f
AFP_ENG_20091015.0364
AFP_ENG_20100130.0284
AFP_ENG_20100423.0583
AFP_ENG_20100505.0537
AFP_ENG_20100630.0660
APW_ENG_20090605.0323
APW_ENG_20090611.0337
APW_ENG_20100508.0084
APW_ENG_20101214.0097
CNA_ENG_20101001.0032
NYT_ENG_20130628.0102
XIN_ENG_20100114.0378
XIN_ENG_20100206.0090
bolt-eng-DF-170-181103-8901874
bolt-eng-DF-170-181103-8908896
bolt-eng-DF-170-181109-48534
bolt-eng-DF-170-181109-60453
bolt-eng-DF-170-181118-8874957
bolt-eng-DF-170-181122-8791540
bolt-eng-DF-170-181122-8793828
bolt-eng-DF-170-181122-8803193
bolt-eng-DF-199-192783-6864512
bolt-eng-DF-199-192909-6666973
bolt-eng-DF-200-192403-6250142
bolt-eng-DF-200-192446-3810246
bolt-eng-DF-200-192446-3810611
bolt-eng-DF-200-192451-5802600
bolt-eng-DF-200-192453-5806585
bolt-eng-DF-203-185933-21070100
bolt-eng-DF-203-185938-398283
bolt-eng-DF-212-191665-3129265
```

ALTA Shared Task papers

Overview of the 2015 ALTA Shared Task:

Identifying French Cognates in English Text

Laurianne Sitbon
Queensland University of Technology (QUT)
l.sitbon@qut.edu.au

Diego Molla
Macquarie University, Sydney, Australia
diego.molla-aliod@mq.edu.au

Haoxing Wang
Queensland University of Technology (QUT)
haoxing.wang@hdr.qut.edu.au

Abstract

This paper presents an overview of the 6th ALTA shared task that ran in 2015. The task was to identify in English texts all the potential cognates from the perspective of the French language. In other words, identify all the words in the English text that would acceptably translate into a similar word in French. We present the motivations for the task, the description of the data and the results of the 4 participating teams. We discuss the results against a baseline and prior work.

1 Introduction

Because many languages have evolved from a shared source language (e.g. Indo-European languages), many words in their vocabularies are the same or are very similar. Additionally, global communications have facilitated the transfer of words from one language to another in modern languages. As a result, when learning a related language, a learner's native language can support the acquisition and understanding of vocabulary words that are identical or similar in both languages.

A vocabulary word is a spelling associated to a particular meaning. Such pairs of identical or similar words that also share meaning across two languages are referred to as cognates. Definitions can vary in the level of similarity (exact or similar spelling, exact or similar pronunciation, or both). So far, research on detecting cognates has focused on being able to identify pairs of cognates in lists of presented pairs of words.

In contrast, in this shared task we use the notion of potential cognate in a target language with reference to a source language: a word in the target language that could be translated by a similar word in the source language such that these words form a cognates pair. Being able to identify these potential cognates in texts could provide technologies to extract easy to understand sentences and could support measures of reading difficulty (Uitdenbogerd, 2005) which can in turn be embedded in ranking information retrieval results or in sentence selection for summarization.

In 2015, the sixth Australasian Language Technology Association (ALTA) shared task was set to identify in English texts all the potential cognates from the perspective of the French language. A total of 6 teams registered to the competition, with 4 teams submitting their results.

In this paper we present some background for the task, describe the dataset and contrast the results of the participants against baselines and previous work. Section 2 presents some background and prior work, Section 3 presents the task, the dataset and the evaluation measures. Section 4 provides the results of the participants. Section 5 discusses the results and future work.

2 Cognates identification and detection

Cognates are pairs of words that are similar in spelling/pronunciation as well as meaning in two languages. By extension, as mentioned above, we refer here to cognates as words in one language that would, in their context of use, acceptably translate into a word in the second language with which they would form a cognate pair.

We also refer here to true cognates as per this definition, as opposed to false cognates (also referred to as false friends) which appear in both languages' lexicons but bear different meanings

Laurianne Sitbon, Diego Mollá and Haoxing Wang. 2015. Overview of the 2015 ALTA Shared Task: Identifying French Cognates in English Text . In *Proceedings of Australasian Language Technology Association Workshop*, pages 134−137.

(such as *pain* in English and *pain* in French (bread)), and as opposed to semi-cognates, which, depending on their context of use, may be either true cognates or false cognates (such as *chair* in English that translates into French as *chaise* if one refers to furniture (false cognate) as *chaire* if one refers to a University position (true cognate), while *chair* in French means *flesh* in English (false cognate)).

The task of detecting potential cognates is in contrast to many experimental settings in the literature that focused on detecting pairs of cognates amongst pairs of words in both languages.

Early work investigated the use of single orthographic or phonetic similarity measures, such as Edit Distance (ED) (Levenshtein, 1966), Dice coefficient (Brew and McKelvie, 1996), Longest Common Subsequence Ratio (LCSR) (Melamed, 1999).

Kondrak and Dorr (2004) reported that a simple average of several orthographic similarity measures outperformed all the measures on the task of the identification of cognates for drug names. More recently, Rama (2014) combined the subsequence features and a number of word shape similarity scores as features to train a SVM model. Kondrak (2001) proposed COGIT, a cognate-identification system that combines phonetic similarity with semantic similarity, the latter being measured from a distance between glosses in a lexical handcrafted resource. Frunza (2006) explored a range of machine learning techniques for word shape similarity measures, and also investigated the use of bi-lingual dictionaries to detect if the words were likely translations of each other. Mulloni, Pekar, Mitkov and Blagoev (2007) also combined orthographic similarity and semantic similarity, the latter being measured based on lists of collocated words.

In previous work, Wang (2014) established an initial version of the dataset proposed in the shared task, and used it to evaluate a new approach. This approach uses word shape similarity measures on pairs selected using word sense disambiguation techniques in order to account for context when seeking possible translations. The implementation is based on BabelNet, a semantic network that incorporates a multilingual encyclopedic dictionary. This work explored a variety of ways to leverage several similarity measures, including thresholds and machine learning.

3 The 2015 ALTA Shared Task

The task of the 2015 ALTA Shared Task was to identify in English texts all the potential cognates from the perspective of the French language. In other words, identify all the words in the English text that would acceptably translate into a similar word in French.

3.1 Dataset

Participants were provided with a training set that is approximately the same size as the testing set. Each set was composed of 30 documents, 5 in each of the following genres: novel, subtitles, sports news, political news, technology news, and cooking recipes. While the separations between the documents was included in both the training and testing data, the categories of documents were not released for the task.

Because we focus on transparency for understanding, we consider similarity (not equality) in either spelling or pronunciation as supporting access to meaning. A single human annotator has identified the potential cognates accordingly.

Similarity: typically similarity is examined at the level of the lemma, so the expected level of similarity would ignore grammatical markers and language-specific suffixes and flexions (for example *negociating* and *negocier* would be considered cognates as the endings that differ respond to equivalent grammatical markers in the languages, similarly for *astrologer* and *astrologue,* or *immediately* and *immediatement*), accented letters are considered equivalent to those without accents and unpronounced letters are ignored (hence *chair* in the French sense *chaire* would be considered true cognate since the *e* at the end is not pronounced). In addition, weak phonetic differences (such as the use of *st* instead of *t* in words such as *arrest* vs. *arrêt,* some vowel substitutions such as *potatoes* vs. *patates)* tend to be ignored and there is more flexibility on long words than on short words.

Rules for proper names: people's names are never considered cognates. Names of companies and products are not considered cognates where the name is a unique word (eg. *Facebook)*, but the words are considered on an individual basis where the name is also a noun compound (eg. in Malaysian Airlines, where *Malaysian* is a cognate, but not *Airlines*). Names of places may be cognates depending to their level of similarity with their translation.

3.2 Task description

The data presented for the task was divided into document text and annotation files. Document text files were formatted with one word (with punctuation attached, if present) per line and each line starts with the line number followed by a space (see Fig.1.a). Document boundaries were indicated by a document id marker.

```
1   <docid 1>
1   Chewy
2   little
3   drops
4   of
5   chocolate
6   cookies,
7   covered
8   with
9   peanuts
```

Figure 1.a Document Text File

```
Eval_id,Cognates_id
1, 6 7
```

Figure 1.b Annotation File

Annotation files were in .csv format. Each line comprised a document number in the first column, and a space delimited list of cognate term indices in the second column.

For instance, to indicate that `chocolate' (index 6) and `cookies' (index 7) are cognates of French words, the annotation file will include the entry shown on Figure 1.b.

Participants were provided with a document text file and corresponding annotation file for the training set, and with a document text file and a sample annotation file (produced by the baseline system, see below) for the test set, and they had to submit their own corresponding annotation file.

3.3 Evaluation

The evaluation measure used for the competition is the mean f-score as defined by the “Kaggle in Class” [1]platform:

$$F1=2pr/(p+r)$$

$$\text{where } p=tp/(tp+fp),\ r=tp/(tp+fn)$$

Where precision (*p*) is the ratio of true positives (tp) to all predicted positives (tp + fp) and recall (*r)* is the ratio of true positives to all actual positives (tp + fn).

However we will discuss the results in terms of recall and precision as well.

3.4 Baselines

The baseline for the task was produced by using a list of 1,765 known English/French cognate words (also matching for singular forms of plurals). Each word in the document text that belonged to the list was deemed to be a cognate for the purpose of the task. As demonstrated in prior work, such baseline tends to yield a high precision but a very low recall.

In addition to the baseline, we ran the task against the system proposed by Wang (2014). The implementation uses BabelNet (Navigli and Ponzetto, 2012) for disambiguating and accessing candidate translations, and integrates 5 measures of similarity (Bi Distance, Dice coefficient, Soundex, Levenshtein, and LCSR) using a Naïve Bayes classifier to assign the cognates labels.

4 Results

The evaluation was performed via the “Kaggle in Class” platform. This platform supports the partition of the test data into a public and a private component. When a team submitted a run, the participants received instant feedback on the results of the public data set, and the results of the private data set was kept for the final ranking. We used the default 50-50 partition provided by Kaggle in Class. The results are reported in Table 1. The table also includes the results returned by the baseline and the system proposed by Wang (2014).

System	Public	Private
LookForward	**0.705**	**0.769**
LittleMonkey	0.671	0.714
Wang(2014)	0.63	0.669
MAC	0.599	0.669
toe_in	0.37	0.367
Baseline	0.229	0.342

Table 1**:** F1 measure results

In Table 2 are presented the results evaluated posterior to the task in terms of recall and precision.

System	Public	Private

[1] http://inclass.kaggle.com/c/alta-2015-challenge/

	R	P	R	P
LookForward	0.76	0.69	**0.79**	0.76
LittleMonkey	0.77	0.62	0.77	0.67
Wang(2014)	**0.81**	0.54	**0.79**	0.60
MAC	0.72	0.54	0.74	0.63
toe_in	0.27	0.62	0.27	0.63
Baseline	0.15	**0.72**	0.22	**0.91**

Table 2**:** recall (R) and precision (P) results

5 Discussion and future work

The rankings between the public and the private test sets are consistent, and therefore the team LookForward is a clear winner. Both LookForward and Little Monkey achieved better results than Wang (2014), and MAC lagged closely behind. The descriptions of the systems used by LookForward, MAC, and toe_in can be found in the proceedings of the ALTA 2015 workshop. Whereas in the teams LookForward and MAC the system used a distance metric that compared the original word with the translation provided by a machine translation system, in the team toe_in the system was based on the intersection of an English and a French lexicon after applying a set of lexical transformations.

As predicted, the baseline had a high precision, and in fact it was the highest of all runs. It is also interesting to observe that the Wang (2014) system is the next highest in recall, while a bit lower in precision. It is important to note that while similar, the annotations on the dataset used in the 2014 paper was slightly different to the one of the 2015 shared task, however the system has not been retrained. This explains a drop in f-measure compared to the results presented in the paper.

Because of a fairly subjective definition of cognates, the annotation of the data can strongly depend on the annotator's personal viewpoint. It would be very interesting to have the dataset re-annotated by 2 more annotators to be able to measure inter-annotator agreement. This would allow judging whether the performance of the best systems reaches the level of humans on the task.

However, in order to put some perspective on the results, it will be even more interesting to measure the impact of the f-measure levels on various tasks such as measuring readability, or selecting sentences or paragraphs in a computer supported language learning system. One could think that a system stronger in precision would be more appropriate to select easy-to-read sentences, while a system stronger in recall may lead to better estimates of reading difficulty.

References

Brew, C., and McKelvie, D. (1996). Word-pair extraction for lexicography. *Proceedings of the second International conference on new methods in language processing,* Ankara, Turkey, 45-55.

Frunza, O.M. (2006). Automatic identification of cognates, false friends, and partial cognates. *Masters Abstracts International*, *45*, 2520.

Kondrak, G. (2001). Identifying cognates by phonetic and semantic similarity. Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies, Pittsburgh, Pennsylvania, USA, 1-8. doi: 10.3115/1073336.1073350.

Kondrak, G., and Dorr, B. (2004*).* Identification of confusable drug names: A new approach and evaluation methodology. *Proceedings of the 20th international conference on Computational Linguistics,* 952-958. doi: 10.3115/1220355.1220492.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet physics doklady, 10,* 707.

Melamed, I.D. (1999). Bitext maps and alignment via pattern recognition. *Comput Linguist., 25*(1), 107-130.

Mulloni, A., Pekar, V., Mitkov, R., & Blagoev, D. (2007). Semantic evidence for automatic identification of cognates. *Proceedings of the 2007 workshop of RANLP: Acquisition and management of multilingual lexicons,* Borovets, Bulgaria , 49-54.

Navigli R. and Ponzetto S. (2012). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network . *Artificial Intelligence*, 193, Elsevier, pp. 217-250

Rama, T. (2014). Gap-weighted subsequences for automatic cognate identification and phylogenetic inference. arXiv: 1408.2359.

Uitdenbogerd, S. (2005). Readability of French as a foreign language and its uses. *Proceedings of the Australian Document Computing Symposium*, 19-25.

Wang, H., Sitbon, S. (2014), Multilingual lexical resources to detect cognates in non-aligned texts, *Proceedings of the Twelfth Annual Workshop of the Australasia Language Technology Association (ALTA)*, Melbourne, Australia, November 2014.

Cognate Identification using Machine Translation

Shervin Malmasi
Centre for Language Technology
Macquarie University
Sydney, NSW, Australia
shervin.malmasi@mq.edu.au

Mark Dras
Centre for Language Technology
Macquarie University
Sydney, NSW, Australia
mark.dras@mq.edu.au

Abstract

In this paper we describe an approach to automatic cognate identification in monolingual texts using machine translation. This system was used as our entry in the 2015 ALTA shared task, achieving an F1-score of 63% on the test set. Our proposed approach takes an input text in a source language and uses statistical machine translation to create a word-aligned parallel text in the target language. A robust measure of string distance, the Jaro-Winkler distance in this case, is then applied to the pairs of aligned words to detect potential cognates. Further extensions to improve the method are also discussed.

1 Introduction

Cognates are words in different languages that have similar forms and meanings, often due to a common linguistic origin from a shared ancestor language.

Cognates play an important role in Second Language Acquisition (SLA), particularly between related languages. However, although they can accelerate vocabulary acquisition, learners also have to be aware of false cognates and partial cognates. False cognates are similar words that have distinct, unrelated meanings. In other cases, there are *partial* cognates: similar words which have a common meaning only in some contexts. For example, the word *police* in French can translate to *police*, *policy* or *font*, depending on the context.

Cognates are a source of learner errors and the detection of their incorrect usage, coupled with correction and contextual feedback, can be of great use in computer-assisted language learning systems. Additionally, cognates are also useful for estimating the readability of a text for non-native readers.

The identification of such cognates have also been tackled by researchers in NLP. English and French are one such pair that have received much attention, potentially because it has been posited that up to 30% of the French vocabulary consists of cognates (LeBlanc and Séguin, 1996).

This paper describes our approach to cognate identification in monolingual texts, relying on statistical machine translation to create parallel texts. Using the English data from the shared task, the aim was to predict which words have French cognates. In §2 we describe some related work in this area, followed by a brief description of the data in §3. Our methodology is described in §4 and results are presented in §5.

2 Related Work

Much of the previous work in this area has relied on parallel corpora and aligned bilingual texts. Such approaches often rely on orthographic similarity between words to identify cognates. This similarity can be quantified using measures such as the edit distance or dice coefficient with n-grams. Brew and McKelvie (1996) applied such orthographic measures to extract English-French cognates from aligned texts.

Phonetic similarity has also been shown to be useful for this task. Kondrak (2001), for example, proposed an approach that also incorporates phonetic cues and applied it to various language pairs.

Semantic similarity information has been employed for this task as well; this can help identify false and partial cognates which can help improve accuracy. Frunza and Inkpen (2010) combine various measures of orthographic similarity using machine learning methods. They also use word senses to perform partial cognate between two languages. All of their methods were applied to English-French. Wang and Sitbon (2014) combined orthographic measures with word sense disambiguation information to consider context.

Shervin Malmasi and Mark Dras. 2015. Cognate Identification using Machine Translation . In *Proceedings of Australasian Language Technology Association Workshop*, pages 138−141.

Cognate information can also be used in other tasks. One example is Native Language Identification (NLI), the task of predicting an author's first language based only on their second language writing (Malmasi and Dras, 2015b; Malmasi and Dras, 2015a; Malmasi and Dras, 2014). Nicolai et al. (2013) developed new features for NLI based on cognate interference and spelling errors. They propose a new feature based on interference from cognates, positing that interference may cause a person to use a cognate from their native language or misspell a cognate under the influence of the L1 version. For each misspelled English word, the most probable intended word is determined using spell-checking software. The translations of this word are then looked up in bilingual English-L1 dictionaries for several of the L1 languages. If the spelling of any of these translations is sufficiently similar to the English version (as determined by the edit distance and a threshold value), then the word is considered to be a cognate from the language with the smallest edit distance. The authors state that although only applying to four of the 11 languages (French, Spanish, German, and Italian), the cognate interference feature improves performance by about 4%. Their best result on the test was 81.73%. While limited by the availability of dictionary resources for the target languages, this is a novel feature with potential for further use in NLI. An important issue to consider is that the authors' current approach is only applicable to languages that use the same script as the target L2, which is Latin and English in this case, and cannot be expanded to other scripts such as Arabic or Korean. The use of phonetic dictionaries may be one potential solution to this obstacle.

3 Data

The data used in this work was provided as part of the shared task. It consists of several English articles divided into an annotated training set (11k tokens) as well as a test set (13k tokens) used for evaluating the shared task.

4 Method

Our methodology is similar to those described in §2, attempting to combine word sense disambiguation with a measure of word similarity. Our proposed method analyzes a monolingual text in a *source* language and identifies potential cognates in a *target* language. The source and target languages in our work are English and French, respectively.

The underlying motivation of our approach is that many of the steps in this task, *e.g.* those required for WSD, are already performed by statistical machine translation systems and can thus be deferred to such a pre-existing component. This allows us to convert the text into an aligned translation followed by the application of word similarity measures for cognate identification. The three steps in our method are described below.

4.1 Sentence Translation

In the first step we translate each sentence in a document. This was done at the sentence-level to ensure that there is enough context information for effectively disambiguating the word senses.[1] It is also a requirement here that the translation include word alignments between the original input and translated text.

For the machine translation component, we employed the Microsoft Translator API.[2] The service is free to use[3] and can be accessed via an HTTP interface, which we found to be adequate for our needs. The Microsoft Translator API can also expose word alignment information for a translation.

We also requested access to the Google Translate API under the University Research Program, but our query went unanswered.

4.2 Word Alignment

After each source sentence has been translated, the alignment information returned by the API is used to create a mapping between the words in the two sentences. An example of such a mapping for a sentence from the test set is shown in Figure 1.

This example shows a number of interesting patterns to note. We see that multiple words in the source can be mapped to a single word in the translation, and vice versa. Additionally, some words in the translation may not be mapped to anything in the original input.

[1] We had initially considered doing this at the phrase-level, but decided against this.

[2] `http://www.microsoft.com/en-us/translator/translatorapi.aspx`

[3] For up to 2m characters of input text per month, which was sufficient for our needs.

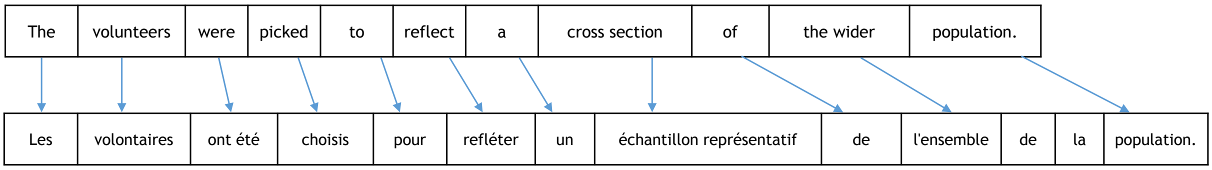


Figure 1: An example of word alignment between a source sentence from the test set (top) and its translation (bottom).

4.3 Word Similarity Comparison Using Jaro-Winkler Distance

In the final step, words in the alignment mappings are compared to identify potential cognates using the word forms.

For this task, we adopt the Jaro-Winkler distance which has been shown to work well for matching short strings (Cohen et al., 2003). This measure is a variation of the Jaro similarity metric (Jaro, 1989; Jaro, 1995) that makes it more robust for cases where the same characters in two strings are positioned within a short distance of each other, for example due to spelling variations. The measure computes a normalized score between 0 and 1 where 0 means no similarity and 1 denotes an exact match.

For each pair of aligned phrases, every word in the source phrases was compared against each word in the aligned phrase to calculate the Jaro-Winkler distance. A minimum treshold of 0.84 was set to detect matches; this value was chosen empirically.

Under some circumstances, such as appearing before a vowel, French articles and determiners may combine with the noun.[4] Accordingly, we added a rule to remove such prefixes (*d'*, *l'*) from translated French words prior to calculating the distance measure. Additionally, all accented letters (*e.g.* *é* and *è*) were replaced with their unaccented equivalents (*e.g.* *e*). We found that these modifications improved our accuracy.

4.4 Evaluation

Evaluation for this task was performed using the the mean F1 score, conducted on a per-token basis. This is a metric based on precision – the ratio of true positives (tp) to predicted positives (tp + fp) – and recall – the ratio of true positives to actual positives (tp + fn). The F1 metric is calculated as:

$$F1 = 2\frac{pr}{p+r} \text{ where } p = \frac{tp}{tp+fp}, \quad r = \frac{tp}{tp+fn}$$

Here p refers to precision and r is a measure of recall.[5] Results that maximize both will receive a higher score since this measure weights both recall and precision equally. It is also the case that average results on both precision and recall will score higher than exceedingly high performance on one measure but not the other.

5 Results and Discussion

Our results on the test set were submitted to the shared task, achieving an F1-score of 0.63 for detecting cognates.[6] The winning entry was 10% higher and scored 0.73.

The key shortcoming of our approach is that we only consider the best translation for detecting cognates. However, a word in the source language may translate to one or more words in the target language, one or more of which could be cognates. However, the cognate(s) may not be the preferred translation chosen by the translation system and therefore they would not be considered by our system.

This was not by design, but rather a technical limitation of the Microsoft Translator API. Although the API provides word alignment information, this is only available for the preferred translation.[7] A separate method is provided for retrieving the n-best translations which could contain relevant synonyms, but it is unable to provide word alignments.

[4] For example, *l'enfant* (the child).

[5] See Grossman (2004) for more information about these metrics.

[6] We obtained F1-scores of 0.67 and 0.59 on the private and public leaderboards, respectively.

[7] Details about this and other restrictions can be found at `https://msdn.microsoft.com/en-us/library/dn198370.aspx`

By using a different machine translation system, one capable of providing alignment information for the n-best translations, our approach could be extended to consider the top n translations. Given the good results using only the preferred translations, this can be considered a very promising direction for additional improvement and is left for future work.

We also noted that there were some idiosyncrasies in the annotation of the training data that were not explicitly outlined. One example is that proper nouns referring to locations, *e.g. Russia, Ukraine* and *Afghanistan*, were annotated whilst other proper nouns were not. Our system would require additional components to distinguish different classes of named entities to be able to implement this logic.

To conclude, we proposed an approach that takes an input text in a source language and uses statistical machine translation to create a word-aligned parallel text in the target language. A robust measure of string distance, the Jaro-Winkler distance in this case, was then applied to the pairs of aligned words to detect potential cognates. The results here are promising and could potentially be further improved using the extensions described in this section.

References

Chris Brew and David McKelvie. 1996. Word-pair extraction for lexicography. In *Proceedings of the 2nd International Conference on New Methods in Language Processing*, pages 45–55.

William Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. A comparison of string metrics for matching names and records. In *KDD workshop on data cleaning and object consolidation*, volume 3, pages 73–78.

Oana Frunza and Diana Inkpen. 2010. Identification and disambiguation of cognates, false friends, and partial cognates using machine learning techniques. *International Journal of Linguistics*, 1(1).

David A Grossman. 2004. *Information retrieval: Algorithms and heuristics*, volume 15. Springer.

Matthew A Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

Matthew A Jaro. 1995. Probabilistic linkage of large public health data files. *Statistics in Medicine*, 14(5-7):491–498.

Grzegorz Kondrak. 2001. Identifying cognates by phonetic and semantic similarity. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

Raymond LeBlanc and Hubert Séguin. 1996. Les congénères homographes et parographes anglaisfrançais. *Twenty-Five Years of Second Language Teaching at the University of Ottawa*, pages 69–91.

Shervin Malmasi and Mark Dras. 2014. Chinese Native Language Identification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL-14)*, pages 95–99, Gothenburg, Sweden, April. Association for Computational Linguistics.

Shervin Malmasi and Mark Dras. 2015a. Large-scale Native Language Identification with Cross-Corpus Evaluation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2015)*, pages 1403–1409, Denver, CO, USA, June. Association for Computational Linguistics.

Shervin Malmasi and Mark Dras. 2015b. Multilingual Native Language Identification. In *Natural Language Engineering*.

Garrett Nicolai, Bradley Hauer, Mohammad Salameh, Lei Yao, and Grzegorz Kondrak. 2013. Cognate and Misspelling Features for Natural Language Identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 140–145, Atlanta, Georgia, June. Association for Computational Linguistics.

Haoxing Wang and Laurianne Sitbon. 2014. Multilingual lexical resources to detect cognates in non-aligned texts. In *Proceedings of the Australasian Language Technology Association Workshop 2014*, volume 12, pages 14–22.

Word Transformation Heuristics Against Lexicons for Cognate Detection

Alexandra L. Uitdenbogerd
RMIT University
GPO Box 2476, Melbourne VIC 3001 Australia
sandrau@rmit.edu.au

Abstract

One of the most common lexical transformations between cognates in French and English is the presence or absence of a terminal "e". However, many other transformations exist, such as a vowel with a circumflex corresponding to the vowel and the letter s. Our algorithms tested the effectiveness of taking the entire English and French lexicons from Treetagger, deaccenting the French lexicon, and taking the intersection of the two. Words shorter than 6 letters were excluded from the list, and a set of lexical transformations were also used prior to intersecting, to increase the potential pool of cognates. The result was 15% above the baseline cognate list in the initial test set, but only 1% above it in the final test set. However, its accuracy was consistant at about 37% for both test sets.

1 Credits

2 Introduction

When assessing readability of English for French native speakers, or French for English native speakers, the cognates — words with similar appearance and meaning — tend to be relatively difficult words, making traditional readability measures less effective than a simple average words per sentence (Uitdenbogerd, 2005). While most words that look similar in French and English are cognates, some commonly occurring words that look similar, such as "a", "as", and "an", tend to be false friends. Other words are partial cognates, having a similar meaning only in some situations (Wang and Sitbon, 2014). Our approach ignored all context and focussed on simple heuristics. All approaches were based on taking the intersection of the French and English lexicons of the Treetagger Part of Speech tagger (Schmid, 1994), after applying a set of lexical transformations. The submission was a "quick and dirty hack" implemented on little sleep during conference attendance, and merely demonstrates that a simple heuristic-based algorithm can beat a manually curated list of cognates, albeit not by much. However, the approach should perform better than demonstrated in the ALTW challenge if applied more comprehensively.

3 Algorithms

The first "quick and dirty" baseline algorithm (**Algorithm 1**) only looks for exact matches once case and accents are preprocessed:

1. Replace the accented letters in the French lexicon with unaccented letters. For example, replace "ê" and "é" with "e".
2. Casefold and remove punctuation from the words in the source and cognate file.
3. Take the intersection of the sanitised source and cognate file words.
4. All words in the text that are in the intersection are counted as cognates.

Algorithm 2 discards any Algorithm 1 words of 5 letters or less as false friends. Common false friend English words that become discarded as a result include: "a", "as", "an". However, the following cognates are also discarded: "ah", "oh".

Algorithm 3 uses lexical transformations to the French lexicon list before intersecting with the English lexicon. It is done with and without plurals. The list is based on observation on reading a French text. While there are many transformation rules, they are not comprehensive. In particular, words in which multiple transformations are required to create an exact match are likely to be missed. Figure 1 shows the regular expression-based substitutions applied to the file.

Alexandra Uitdenbogerd. 2015. Word Transformation Heuristics Agains Lexicons for Cognate Detection . In *Proceedings of Australasian Language Technology Association Workshop*, pages 142−144.

Table 1: Training Set Statistics

	Actual	Algorithm 3
Cognates	1670	703
Non-Cognates	9425	10392
Total Words	11095	
Proportion of Cognates	0.150	0.063
Precision		0.661
Recall		0.278
F1		0.390

Table 2: ALTW Challenge 2015 Results

Team	Public	Private
LookForward	0.70478	0.77001
Little Monkey	0.67118	0.71415
MAC	0.59927	0.66857
toe_in (Alg. 4 lex.tr.-shrt)	0.37019	0.36697
Alg. 3 (lex. trans.)	0.31394	0.37272
Alg. 2 (Exact - shrtwrds)	0.23583	0.23908
Baseline	0.22951	0.34158
Alg. 1 (Exact Match)	0.22107	0.27406
Stemmed lexicon match	0.11347	0.14116

Algorithm 4 combines Algorithm 3's lexical transformations with discarding words that are 5 letters or fewer in length.

We also tried stemming but the result was half the accuracy of the original baseline. The final submission used the transformations as well as discarding words of 5 letters or less.

4 Results

Table 1 shows the precision, recall and F1 measure for the training data set.

Table 2 shows the overall results for the ALTW challenge. As can be seen, our entry (toe_in) had the most consistent performance across the two test sets. Of our submissions, Algorithm 3 performed the best on the public test dataset. The best private data submission was a version of Algorithm 3 that didn't discard short words.

In a post-analysis using the training data set we looked at the effect of varying the minimum word length for cognates, holding the base word list constant. Table 3 shows the effect on precision, recall and F measure. Precision increases as the minimum word length is increased, and recall decreases. The sweet spot in the training data set is to discard words that are 4 letters long or less.

Table 3: The effect of minimum word length on cognate detection reliability

Min Length	Precision	Recall	F measure
3	.457	.346	.393
4	.457	.346	.393
5	.579	.323	.414
6	.658	.266	.378
7	.711	.198	.309

5 Discussion

The experimental results demonstrated that a lexically transformed French lexicon intersected with an English lexicon with the shortest words discarded can be a substitute for a manually curated list of cognates, achieving 1 to 15% higher accuracy on the given test sets. A more comprehensive set of lexical transformations is likely to give a slightly higher accuracy again.

However, as the ALTW challenge results demonstrate, this context-free, heuristic approach has only about half the accuracy of the best technique.

Acknowledgments

Some text processing code was written by Aidan Martin.

Support for this project has been provided by the Australian Government Office for Learning and Teaching. The views in this project do not necessarily reflect the views of the Australian Government Office for Learning and Teaching.

References

A. L. Uitdenbogerd. 2005. Readability of French as a foreign language and its uses. In A. Turpin and R. Wilkinson, editors, *Australasian Document Computing Symposium*, volume 10, December.

H. Schmid. 1994. Probabilitic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*.

H. Wang and L. Sitbon. 2014. Multilingual lexical resources to detect cognates in non-aligned texts. In G. Ferraro and S. Wan, editors, *Proceedings of the Australasian Language Technology Association Workshop*, pages 14–22, Melbourne, Victoria, Australia, November. ALTA.

```
grep "e$" $1 | sed "s/e$//"
grep "ait$" $1 | sed "s/ait$/ed/"
grep "aient$" $1 | sed "s/aient$/ed/"
grep "gue$" $1 | sed "s/gue/g/"
grep "é$" $1 | sed "s/é$/y/"
grep "euse$" $1 | sed "s/euse/ous/"
grep "eux$" $1 | sed "s/eux/ous/"
grep "ique$" $1 | sed "s/ique/ic/"
grep "^dé$" $1 | sed "s/^dé/dis/"
grep "ont$" $1 | sed "s/ont$/ount/"
grep "ond$" $1 | sed "s/ond$/ound/"
grep "ant$" $1 | sed "s/ant$/ing/"
grep "ain$" $1 | sed "s/ain$/an/"
grep "aine$" $1 | sed "s/aine/an/"
grep "re$" $1 | sed "s/re$/er/"
grep "ment$" $1 | sed "s/ment$/ly/"
grep "é$" $1 | sed "s/é$/ated/"
grep "é$" $1 | sed "s/é$/ed/"
grep "ée$" $1 | sed "s/ée$/ated/"
grep "ée$" $1 | sed "s/ée$/ed/"
grep "i$" $1 | sed "s/i$/ished/"
grep "ir$" $1 | sed "s/ir$/ish/"
grep "er$" $1 | sed "s/er$/e/"
grep "ô" $1 | sed "s/ô/os/"
grep "ê" $1 | sed "s/ê/es/"
grep "î" $1 | sed "s/î/is/"
grep "ement$" $1 | sed "s/ement$/ly/"
grep "eusement$ $1| sed "s/eusement$/ously/"
grep "isme$" $1 | sed "s/isme$/ism/"
grep "if$" $1| sed "s/if$/ive/"
grep "asse$" $1 | sed "s/asse$/ace/"
grep "eur$" $1 | sed "s/eur$/or/"
grep "eur$" $1 | sed "s/eur$/er/"
grep "eur$" $1 | sed "s/eur$/our/"
grep "^é" $1 | sed "s/^é/es/"
grep "^é" $1 | sed "s/^é/s/"
grep "oût" $1 | sed "s/oût/ost/"
grep "^av" $1 | sed "s/^av/adv"
grep "^aj" $1 | sed "s/^aj/adj"
grep "elle$ $1 | sed "s/elle$/al/"
grep "ette$" $1 | sed "s/ette$/et/"
grep "onne$" $1 | sed "s/onne$/on/"
grep "quer$" $1 | sed "s/quer$/cate/"
grep "ai" $1 | sed "s/ai/ea/"
grep "^en" $1 | sed "s/^en/in/"
grep "ier$" $1 | sed "s/ier$/er/"
```

Figure 1: The set of lexical transformations applied to the French lexicon prior to intersection with the English lexicon. ”$1” is the file containing the French lexicon.

Detecting English-French Cognates Using Orthographic Edit Distance

Qiongkai Xu[1,2], **Albert Chen**[1], **Chang Li**[1]
[1]The Australian National University, College of Engineering and Computer Science
[2]National ICT Australia, Canberra Research Lab
{xuqiongkai, u5708995, spacegoing}@gmail.com

Abstract

Identification of cognates is an important component of computer assisted second language learning systems. We present a simple rule-based system to recognize cognates in English text from the perspective of the French language. At the core of our system is a novel similarity measure, orthographic edit distance, which incorporates orthographic information into string edit distance to compute the similarity between pairs of words from different languages. As a result, our system achieved the best results in the ALTA 2015 shared task.

1 Introduction

Cognates words are word pairs that are similar in meaning, spelling and pronunciation between two languages. For example, "age" in English and "$\hat{a}ge$" in French are orthographically similar while ”father” in English and ”Vater” in German are phonetically similar. There are three types of cognates: true cognates, false cognates and semi-cognates. True cognates may have similar spelling or pronunciation but they are mutual translations in any context. False cognates are orthographically similar but have totally different meanings. Semi-cognates are words that have the same meaning in some circumstances but a different meaning in other circumstances. Finding cognates can help second language learners leverage their background knowledge in their first language, thus improving their comprehension and expanding their vocabulary.

In this paper, we propose an automatic method to identify cognates in English and French with the help of the Google Translator API[1]. Our method calculates the similarity of two words based solely on the sequences of characters involved. After exploring n-gram similarity and edit distance similarity, we propose an orthographic edit distance similarity measure which leverages orthographic information from source language to target language. Our approach achieved first place in the ALTA 2015 shared task.

2 Related Work

There are many ways to measure the similarity of words from different languages. Most popular ones are surface string based similarity, i.e. n-gram similarity and edit distance. An n-gram is a contiguous sequence of n items, normally letters, from a given sequence. There are many popular measures that use n-grams such as DICE (Brew et al., 1996), which uses bi-grams, and Longest Common Subsequence Ratio (LCSR) (Melamed, 1999). LCSR was later found to be a special case of n-gram similarity by Kondrack (Kondrak, 2005), who developed a general n-gram framework. He provided formal, recursive definitions of n-gram similarity and distance, together with efficient algorithms for computing them. He also proved that in many cases, using bi-grams is more efficient than using other n-gram methods. Since LCSR is only a tri-gram measure, using bi-gram similarity and distance can easily outperform LCSR in many cases.

Instead of computing common n-grams, word similarity can be also measured using edit distance. The edit distance between two strings is the minimum number of operations that are needed to transform one string into another. When calculating the edit distance, normally three operations are considered: removal of a single character, insertion of a single character and substitution of one character with another one. Levenshtein defined each of these operations as having unit cost except for substitution (Levenshtein, 1966). Other suggestions have been made to add more opera-

[1] https://code.google.com/p/google-api-translate-java/

Language	Sentence
English	We have to do it out of respect.
French	Nous devons le faire par respect

Table 1: Phrase alignment of machine translation.

tions like merge and split operations in order to consider adjacent characters (Schulz and Mihov, 2002). The algorithm was improved by Ukkonen using a dynamic programming table around its diagonal making it linear in time complexity (Ukkonen, 1985).

3 System Framework

To tackle the ALTA 2015 shared task[2], we propose a system consisting of the following steps:

- Step 1: Translate source words (English) into target words (French). Filter out meaningless words or parts of words.
- Step 2: Calculate the similarity score of all word pairs. Search the best threshold and decide if word pairs are cognates.

3.1 Cognate Candidates Generation

Since there is no aligned French corpus provided in this task, we need to generate cognate candidates by using a machine translator. One approach is to translate English sentences into French sentences followed by extracting the aligned words. Although this approach makes use of the words' context, its quality depends on both the quality of the translator and the word alignment technology. Table 1 shows an example of machine translation and phrase alignment results. We find that "do" (faire) and "it" (le) are in a different order when translated into French. We work around this by translating each sentence word by word using the Google Translator API. A benefit of this approach is that we can cache the translation result of each word, making the system more efficient. The total time of calling the translator API is reduced from more than 22,000 to less than 5,600 in the training and testing sets.

Due to the differences between French and English, an English word (a space-separated sequence of characters) may be translated to more than one word in French. For example, Google Translator translates "language's" to "la langue de". To facilitate the process of determining whether "language's" is a cognate in French and English, we first filter out the "'s" from the English word and the "la" and the "de" from the translation. We can then calculate the similarity of "language" and "langue". More generally, we filter out the definite articles "le", "la" and "les" and the preposition "de" from the phrase given by the translator.

SL:	$len(S) - n + 1$
Max:	$\max\{len(S) - n + 1, len(T) - n + 1\}$
Sqrt:	$\sqrt{(len(S) - n + 1)(len(T) - n + 1)}$

Table 2: Normalization factor for n-gram similarity.

SL:	$len(S)$
Max:	$\max\{len(S), len(T)\}$
Sqrt:	$\sqrt{len(S)len(T)}$

Table 3: Normalization factor for edit distance similarity.

3.2 N-gram and Edit Distance

For character-level n-gram distance, we calculate the number of common n-gram sequences in source S and target T and then divide by L (the normalization factor) to obtain the normalized n-gram distance similarity:

$$n\_sim(S, T) = \frac{|n\text{-}gram(S) \cap n\text{-}gram(T)|}{L}.$$

We consider three candidates for L: source length (SL), maximum length of S and T (Max), and geometric mean of S and T length (Sqrt) (Table 2).

We calculate the edit distance (Levenshtein distance), from $S = \{s_1, s_2, \ldots, s_n\}$ to $T = \{t_1, t_2, \ldots, t_m\}$ using dynamic programming. The following recursion is used:

$$d_{i,j} = \begin{cases} d_{i-1,j-1} & \text{if } s_i = t_j \\ \min\{d_{i-1,j}, d_{i,j-1}\} & \text{if } s_i \neq t_j \end{cases}$$

where $d_{i,j}$ is the edit distance from $s_{1,i}$ to $t_{1,j}$. Then the similarity score is

$$l\_sim(S, T) = 1 - \frac{d_{n,m}}{L}$$

[2] http://www.alta.asn.au/events/alta2015/task.html

where L is the normalization factor. Again, we consider three values for L: SL, Max, Sqrt (Table 3).

Instead of using a machine learning algorithm to determine word similarity, we focus on the most promising feature which is edit distance similarity. We further explore this approach and propose a novel similarity measure. A grid search algorithm is utilized to find the best threshold for our system and which works efficiently.

3.3 Edit Distance with Orthographic Heuristic Rules

Although traditional edit distance similarity can figure out cognates in most cases, orthographic information is not utilized properly. We propose an orthographic edit distance similarity which is used to measure the similarity of each pair. We first generate a map that associates common English pieces to French pieces and allows us to ignore diacritics. Suffixes like "k" and "que" are often a feature of cognates in English and French (e.g. "disk" and "disque"). Mapping "e" to "é", "è" and "ê" helps in finding "system" (English) and "système" (French) as cognates (the accents affect the pronunciation of the word).

If the characters are the same in the two words, the edit distance is zero. Otherwise, we add a penalty, $\alpha \in [0,1]$, to the edit distance if the suffix of length k of the first i characters of the English word maps to the suffix of length l of the first j characters of the French word. α is set to 0.3 according to our experimentation.

$$d_{i,j} = \min \begin{cases} d_{i-1,j-1} & \text{if } s_i = t_j \\ d_{i-k,j-l} + \alpha & \text{if } \{s_{i-k+1}, \ldots, s_i\} \\ & \rightarrow \{t_{j-l+1}, \ldots, t_j\} \\ \{d_{i-1,j}, d_{i,j-1}\} & \text{elsewhere} \end{cases}$$

All orthographic heuristic rules (map) are illustrated in Table 4.

$$\mathit{e\_sim}(S,T) = 1 - \frac{d_{n,m}}{L}$$

The normalization factor is the same as the one used in Section 3.2. The pseudocode for calculating the orthographic edit distance is provided in Algorithm 1.

English	French
e	é è ê ë
a	â à
c	ç
i	î ï
o	ô
u	û ù ü
k	que

Table 4: English-French orthographic Heuristic Rules for orthographic edit distance.

L	Precision(%)	Recall(%)	F-1(%)
SL	73.21	76.59	74.86
Max	72.40	**79.94**	75.98
Sqrt	**75.06**	77.31	**76.17**

Table 5: Result of bi-gram similarity on training dataset using different normalization methods.

4 Experiments and Results

4.1 Dataset and Evaluation

The ALTA 2015 shared task is to identify all words in English texts from the perspective of the French language. Training data are provided, while labels of test data are not given. Since our system only focuses on limited similarity measurements, we believe a development set is not necessary. For each approach discussed, we use the training data to find the best threshold. Then, we test our system on the public testing data. If the results improve in both training and public testing, we submit our system.

The evaluation metric for this competition is F_1 score, which is commonly used in natural language processing and information retrieval tasks. Precision is the ratio of true positives (tp) to all predicted positives (tp+fp). Recall is the ratio of true positives (tp) to all actual positive samples (tp+fn).

$$P = \frac{tp}{tp + fp}, \; R = \frac{tp}{tp + fn}.$$

$$F_1 = 2\frac{P \cdot R}{P + R}$$

4.2 Experiment Results

We first compare bi-gram similarity and traditional edit distance similarity (Tables 5 and 6). SL, Max and Sqrt are all tested as normalization

Algorithm 1 Orthographic Edit Distance

```
function ORTHEDITDIST($s$, $t$, $map$)
    $sl \leftarrow len(s)$
    $tl \leftarrow len(t)$
    for $i \leftarrow 0$ to $sl$ do
        $d[i][0] \leftarrow i$
    end for
    for $j \leftarrow 0$ to $tl$ do
        $d[0][j] \leftarrow j$
    end for
    for $i \leftarrow 0$ to $sl - 1$ do
        for $j \leftarrow 0$ to $tl - 1$ do
            $d[i+1][j+1] \leftarrow \min\{d[i+1][j]+1, d[i][j+1]+1, d[i+1][j+1]+1\}$
            for each orthographic pair $(s', t')$ in $map$ do
                $i' \leftarrow i - len(s')$
                $j' \leftarrow j - len(t')$
                if $i' \geq 0$ and $j' \geq 0$ then
                    continue
                end if
                if $s.substring(i', i+1) = s'$ and $t.substring(j', j+1) = t'$ then
                    $d[i+1][j+1] \leftarrow \min\{d[i+1][j+1], d[i'][j'] + \alpha\}$
                end if
            end for
        end for
    end for
    return $d[sl][tl]$
end function
```

L	Precision(%)	Recall(%)	F-1(%)
SL	72.49	79.52	75.84
Max	71.80	**80.96**	76.10
Sqrt	**75.23**	78.20	**76.68**

Table 6: Result of edit distance similarity on training dataset using different normalization methods.

L	Precision(%)	Recall(%)	F-1(%)
SL	77.56	75.15	76.34
Max	**75.48**	79.46	**77.42**
Sqrt	74.80	**79.82**	77.23

Table 7: Result of orthographic edit distance similarity on training dataset using different normalization methods.

factors for both approaches. Edit distance similarity constantly outperforms bi-gram similarity (around 0.5% to 1% higher). Orthographic edit distance similarity further improves the result by about 0.5%. Another trend is that Max and Sqrt normalization is better than SL, which only considers the length of source string. Max and Sqrt are competitive to some extent.

According to the previous experiment, we use orthographic edit distance similarity to measure the similarity of words. The maximum length of source word and target word is used as the normalization factor. Using the grid search algorithm, the threshold is set to 0.50. The final F_1 scores on public and private test data are 70.48% and 77.00%, both of which are at top place.

5 Conclusions

We used a translator and string similarity measures to approach the ALTA 2015 shared task, which was to detect cognates in English texts from the respect of French. By using our novel similarity method, orthographic edit distance similarity, our system produced top results in both public and private tests.

Acknowledgement

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

Chris Brew, David McKelvie, et al. 1996. Word-pair extraction for lexicography. In *Proceedings of the 2nd International Conference on New Methods in Language Processing*, pages 45–55. Citeseer.

Grzegorz Kondrak. 2005. N-gram similarity and distance. In *String processing and information retrieval*, pages 115–126. Springer.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

I Dan Melamed. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1):107–130.

Klaus U Schulz and Stoyan Mihov. 2002. Fast string correction with levenshtein automata. *International Journal on Document Analysis and Recognition*, 5(1):67–85.

Esko Ukkonen. 1985. Algorithms for approximate string matching. *Information and control*, 64(1):100–118.