Internship Report:

Sequence Labelling using Distributional Semantic Vectors and Conditional Random Fields

Melanie Tosik
Department Linguistics
University of Potsdam
Potsdam, Germany
tosik@uni-potsdam.de

Directed by: Carsten Lygteskov Hansen

at

textkernel

Amsterdam, The Netherlands

June – August 2014

Abstract

Sequence Labelling using Distributional Semantic Vectors and Conditional Random Fields

Melanie Tosik

The aim of this report is to outline the research that has been carried out during a three-month summer internship at Textkernel in Amsterdam, The Netherlands.

The general objective of the project was to improve the resume parsing model for German. By developing a novel approach to information extraction using sequence labelling, we obtain promising results indicating significant improvements over the current baseline model. In addition to project realisation and overall findings, professional and personal experiences are presented in the course of this report.

Directed by: Carsten Lygteskov Hansen

Acknowledgements

I would like to thank everyone at Textkernel; you have made my three months of training a worthwhile and enlightening experience.

Special gratitude extends to Carsten Lygteskov Hansen for his guidance throughout the internship, for his patience, his enthusiasm and his availability.

My personal thanks are directed towards Mihai Rotaru, not only for proof–reading this report, but for keeping sight of the big picture and the goals and values behind the chosen tools and methods.

Particular thanks also go to the rest of the *Textractor* team: you provided the necessary ideas to make this research project a success.

Contents

About

1.1 Textkernel

In 2001, Textkernel emerged as a commercial R&D spin-off of research in natural language processing and machine learning at the universities of Tilburg, Antwerp and Amsterdam, and quickly developed into a cutting-edge software company providing multilingual recruitment technology in 16 languages for more than 1000 organisations worldwide.

Today, it is the biggest artificial intelligence lab of the European HR sector, employing over 50 experts in language technology and software engineering specialised in information extraction, document understanding, web mining, and semantic searching and matching.

During the internship, we aimed at the development of the *Extract! CV parsing* software.

The *Textractor* team behind it currently consists of six research engineers, mainly concerned with the automatic extraction of relevant information from Curriculum Vitæs (resumes and profiles of business-oriented social networking platforms like LinkedIn or Xing) and job vacancies. Together with semantic job matching, the extraction services process around 10 million documents per year.

1.2 Internship Objective

The overall goal of the internship was to explore new methods of improving CV parsing for German documents. The enhancements should thereby become generalisable to all languages, advancing the production system as a whole.

Within three months, we experimented with a novel approach that integrates distributional semantic word vectors as features for a probabilistic Conditional Random Field (CRF) model performing the task of segmenting and labelling the sequenced data.

Since the infrastructure for the new model architecture had been provided and proven for a few test runs on English CV data, we focused on answering the research question by starting with an extended investigation of the German phrase models.

Project Realisation

2.1 Pipeline Architecture

To understand the focus and the outcome of the work put on display, it is important to have an overview of the data model at large.

Given that in this case the task is to extract structured information in the form of particular phrases like *name* or *address*, the pipeline architecture is designed as illustrated in Figure 2.1.

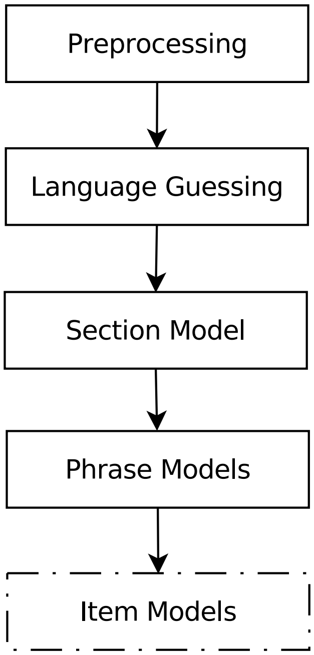


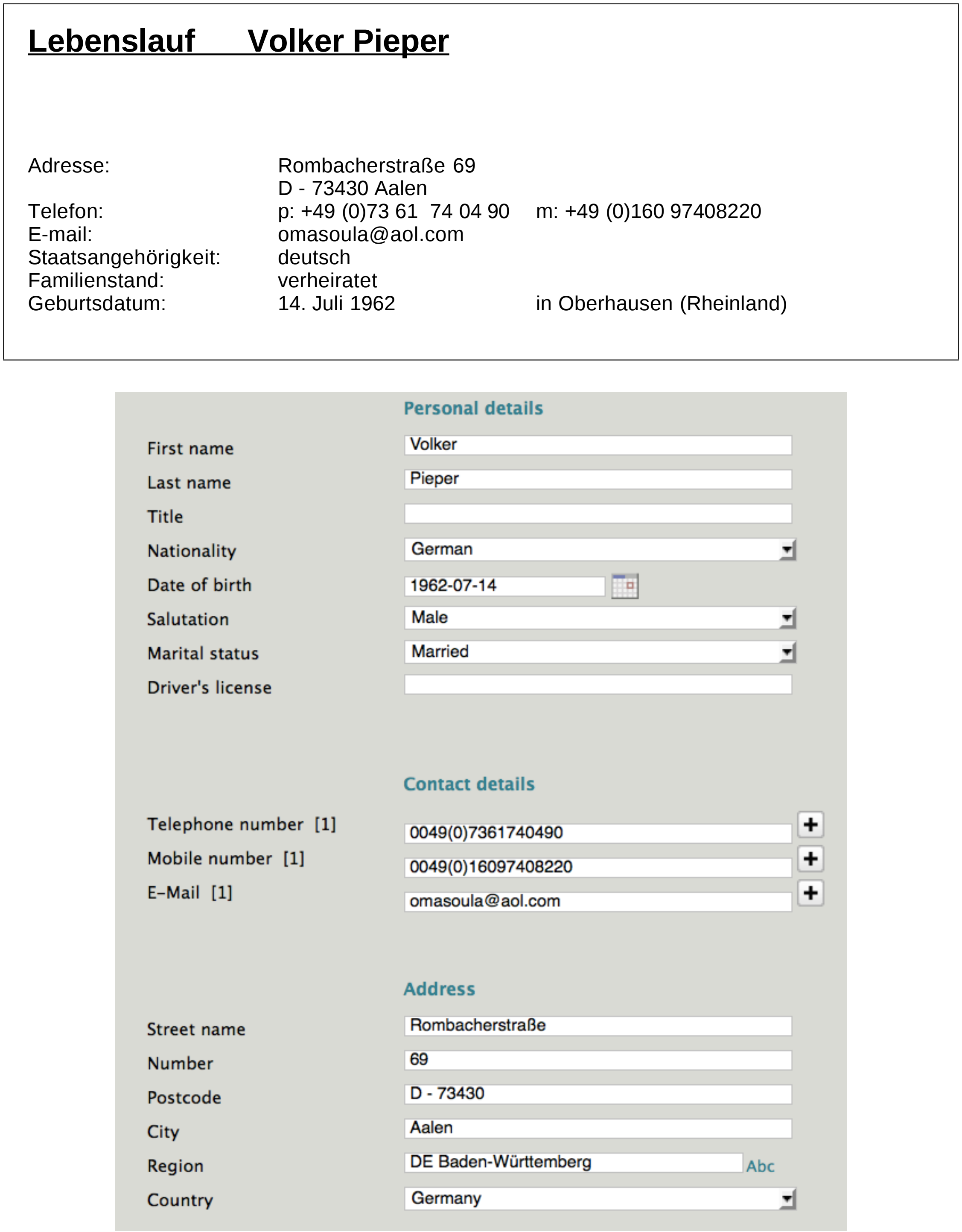
Figure 2.1: Phrase extraction pipeline architecture.

Following the preprocessing of each CV, the language of its content is determined in order to apply the appropriate language models in a cascaded fashion.

Afterwards, the language–specific section model segments the document into the following sections: Cover Letter Section, Personal Section, Education Section, Experience Section, Skills Section and Extracurricular Section. Not all sections are present in every document.

Phrase models then extract the detailed information from the corresponding section.

A working example of personal phrase extraction for German is illustrated in Figure 2.2.



Lebenslauf Volker Pieper

Adresse:	Rombacherstraße 69 D - 73430 Aalen	
Telefon:	p: +49 (0)73 61 74 04 90	m: +49 (0)160 97408220
E-mail:	omasoula@aol.com	
Staatsangehörigkeit:	deutsch	
Familienstand:	verheiratet	
Geburtsdatum:	14. Juli 1962	in Oberhausen (Rheinland)

Personal details

First name	Volker
Last name	Pieper
Title	
Nationality	German
Date of birth	1962-07-14
Salutation	Male
Marital status	Married
Driver's license	

Contact details

Telephone number [1]	0049(0)7361740490
Mobile number [1]	0049(0)16097408220
E-Mail [1]	omasoula@aol.com

Address

Street name	Rombacherstraße
Number	69
Postcode	D - 73430
City	Aalen
Region	DE Baden-Württemberg
Country	Germany

Figure 2.2: Personal section of an example CV and phrase extraction within the *Sourcebox* API.

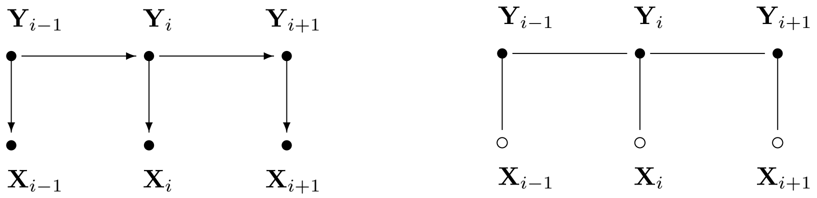


Figure 2.3: Graphical structures of HMMs (left) and chain-structured CRFs (right) for sequences.

An overview of selected sections with extracted fields can be found in Appendix A.1.

For the purpose of improving the phrase models, we disabled section extraction to avoid errors arising from a previous task and worked on gold annotated sections only.

2.2 Sequence Labelling using Conditional Random Fields

Customarily, *Extract! CV Parsing* relies on Trigrams'n'Tags (TnT) [1], an implementation of a Hidden Markov Model (HMM) originally applied to Part–of–Speech (POS) Tagging. In a nutshell, the model assigns a joint probability to paired observation and label sequences and attempts to maximise the joint likelihood of training examples.

However, a generative model like this is not able to account for multiple interacting features or long-range dependencies. Additionally, the handling of unknown words quickly becomes a problem since the lexical probabilities of words which are not present in the lexicon can only be estimated by incorporating some other source.

To overcome these fundamental limitations, we turn to an alternative framework presented in [2]. Instead of modelling fixed observations, a framework based on Conditional Random Fields (CRF) models the joint probability of the entire sequence of labels given the observation sequence. While the generative model relies on word entities and their correspondent part-of-speech tags, the CRF model uses word vector representations (cf. Section 2.3) and a number of different, optional features whose weights can be traded off against each other.

In contrast to the HMM, transition probabilities no longer depend on the previous observation only, but on all available past and future observations as well. Consequently, the CRF model architecture does not impose the strict independence assumptions that the HMM model involves. Because the CRF model can take any possible feature into account, it is additionally viable to integrate high-dimensional vector representations as features of the model. It therefore covers a much larger number of words that would not be represented in the lexicon of the HMM otherwise.

The critical difference between HMM and CRF is illustrated in Figure 2.3 (adapted from [2]). Open circles indicate variables not generated by the model.

For our experiments we use *CRFsuite*[1], an implementation of Conditional Random Fields for labelling sequential data provided by [5].

[1] `http://www.chokkan.org/software/crfsuite/`

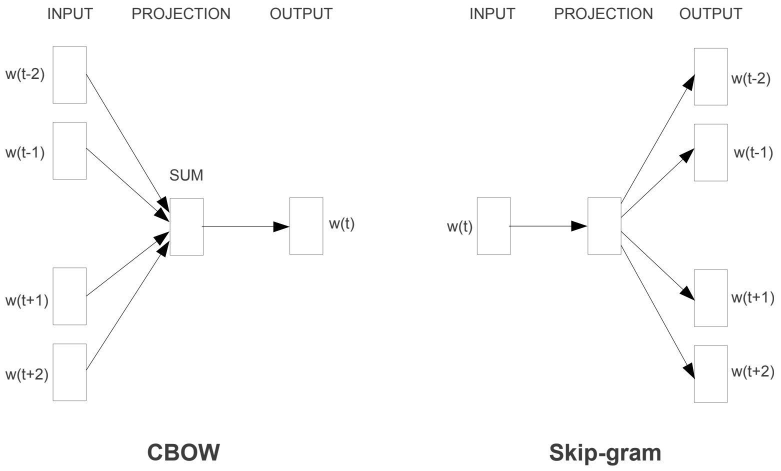


Figure 2.4: CBOW and Skip-gram architecture.

Since the software implements several training methods, we chose an appropriate learning algorithm based on accuracy[2] and the amount of time it takes to train the model. By default, *CRFsuite* uses Limited-memory BFGS[3] (L–BFGS) [4]. We switched to Stochastic Gradient Descent (SGD) [7] after we obtained similar results in only half of the time.

2.3 Distributional Semantic Vectors

As described above, the CRF model learns based on a number of predefined features. The standard configuration includes features that account for the beginning and end of a line, unknown words, high frequency tokens, digits, single characters, multi spaces, capitalisation, as well as the first and last token of each line. When the model is trained on annotated training data, a binary decision is made as to whether the features are present for the current observation or not.

While all of these basic features are useful to recognise patterns in both the layout and the content of the CV, crucial information is added by integrating a distributional semantic vector for each word instead of just the word itself.

Following [3], we use the Skip-gram model to compute continuous vector representations of words learned by a neural network. Unlike the continuous bag-of-words model (CBOW), which predicts a current word based on the context, the Skip-gram model classifies the current word based on another word in the same sentence. By predicting surrounding words within a certain range before and after the current word, the Skip-gram model yields better results than the computationally less expensive CBOW approach.

Both architectures are illustrated in Figure 2.4 taken from [3].

[2] We evaluate using the common recall, precision and F1–Score metrics (character based).

[3] BFGS refers to the Broyden–Fletcher–Goldfarb–Shanno algorithm.

Data source	Number of documents	Number of tokens	Type of (German) data
Original CV sample	13k	23.7M	Sample CVs
German Wikipedia	1.75M	538M	Wikipedia (19–06–2014)
External vacancies	120k	41M	Vacancies
Internal vacancies	200k	86M	Vacancies
New CV sample	200k	145.5M	Sample CVs
Combined sources	2.3M	833M	All data sets combined

Table 2.1: Overview of datasets, their size and the type of data.

For all our experiments, we use the open source *word2vec*[4] toolkit [3].

To verify the process of randomly sampling the size of the context window from a certain range, we regenerate distributional semantic vectors from the same dataset in several independent runs. When using the default value of a maximum skip length of 5 between words, this randomisation does not affect the quality of the vectors beyond introducing a negligible amount of noise.

In order to obtain the best vector representations for the specific task of labelling phrases within certain sections (e.g. *name* and *address* for the personal section), we conduct a number of experiments in which we test different settings for various parameters. The quality of the vectors is measured in terms of the performance of the phrase models and thus strongly dependent on the properties of the current production system.

2.3.1 Data source and amount of data

Data sources and information about size and type of the data are listed in Table 2.1.

Overall best vector representations for the phrase extraction tasks are generated from all the data combined. Training the neural network on Wikipedia works equally well as using the original batch of sample CVs, an important insight for the opening up of new markets for which large amounts of training data are not yet available. Using vacancies (from either external or internal sources) works as well as combining the original CV data with the Wikipedia data. Enriching the original CV data with a larger sample of CV data steadily increases the accuracy of the model. A plateau has not yet been reached.

To test the amount of data needed to get reasonable representations, we use the original batch of sample CVs and experiment with 10%, 33%, 66%, 300%, 600% and 1000% of data. Data sets are generated by either randomly downsampling or duplicating the original set accordingly. While duplicating the data does not improve the initial word vectors, decreasing the amount of CV data has surprisingly little effect: although the decline is incremental, it only amounts to a ~3% drop for generating vectors from 10% of the data.

[4] https://code.google.com/p/word2vec/

2.3.2 Vector size

[3] indicate that a larger dimensionality is beneficial when training on large data sets and report the best results for word pair relationships for a vector size of 300 dimensions. Our experiments evaluate results for 15, 50, 100, 300 and 450 dimensions on the original batch of sample CVs. The baseline parameter is set to 150 dimensions. We find that a higher dimensionality of the vector space does not improve the performance of the phrase models. Best results are obtained by using 150 dimensions, additionally significantly reducing the complexity of the computation.

2.3.3 External dataset

Apart from evaluating the vector representations on the CV parsing task, we apply them to solve the semantic relatedness task on an external dataset to validate quality and generalisability of the induced vector space.

The *Gur350*[5] dataset for German contains 350 word pairs (involving nouns, verbs and adjectives) along with their relatedness scores assigned on a discrete 0–4 scale by 8 subjects with an inter–annotator agreement of 0.69.

We implement cosine similarity and evaluate the word vectors generated from the original sample CVs, yielding a Spearman's rank correlation of 0.23 with 146 unknown words for which we can not induce a vector representation.

Since the dataset consists of 350 non-domain specific word pairs that word vectors generated from CV data do not cover in many cases, we proceed to learn them from the whole German Wikipedia instead, reporting a Spearman's correlation of 0.50 (and 27 unknown words). This value is higher than any of the results[6] presented in [8] but might be due to the steady growth of the encyclopaedia. If we combine original CV and Wikipedia data, we obtain a Spearman's rank correlation of 0.52.

2.4 Phrase Models

2.4.1 German

To get an idea about the expressiveness of the two models and the kinds of errors they make, we started on the PERSONAL PHRASE MODEL[7] for German by comparing the extracted *name* entities against the gold annotated entities. The pattern that we observed suggested that the TnT model generally has a higher recall but over generates by including non-word characters or other strings that are not actually targeted, whereas the CRF model has a higher precision but fails to extract a considerable number of entities at all.

[5] https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-relatedness-datasets/

[6] The highest score using Wikipedia was $\rho = 0.42$ with a similarity measure following [6].

[7] Personal phrase model refers to the phrase extraction model for the personal section.

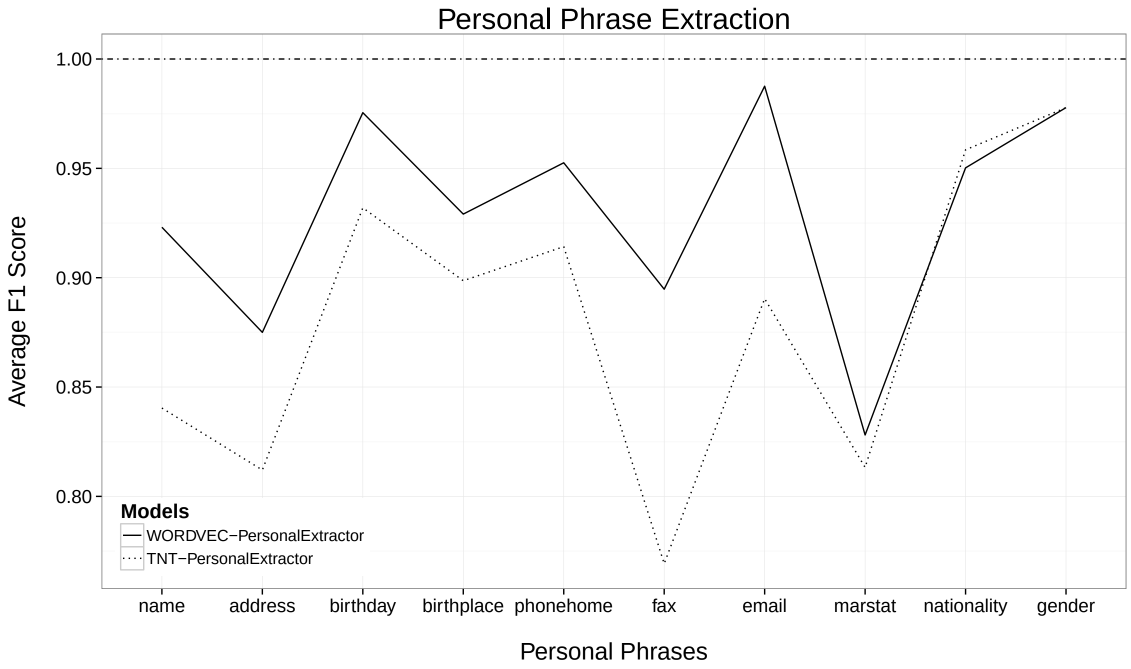


Figure 2.5: Results obtained by the TNT baseline model and the current candidate model.

From this analysis we hypothesised that the low recall for the CRF model might be caused by a large amount of unknown words in the vector space. We investigated the frequency distribution throughout the training data for both models and found 5% of unknown words for the CRF and 11% of unknown words for the TnT model. Moreover, we noticed a confusion between address suffixes, and German names and months[8].

We implemented additional features for street names and months and added gazetteers containing male and female first names from multiple languages. It is still unclear what exactly it is that the HMM–based model grasps but the CRF model utilising vector representations does not yet capture. However, as illustrated in Figure 2.5, the CRF model significantly outperforms the TnT model on all the relevant fields extracted from personal sections.

For the EXPERIENCE PHRASE MODEL, we began with conducting a detailed error analysis on the target of *experience*, denoting precise job titles.

After a substantial amount of flaws in the gold annotations impeded the evaluation, we developed detailed annotation guidelines on how to correct them and fixed the training, development and test partition with the help of the *Data and Quality* team.

Contrary to our expectations, correcting the most frequent types of annotation errors did not improve the performance of the CRF model. However, another interesting property of CRF that has not been mentioned yet is the amount of data required to train the model in the first place. While the performance of the TnT model is strongly dependent on a large

[8] For instance, *April*, *Juli* and *August* can be either first names or months in German.

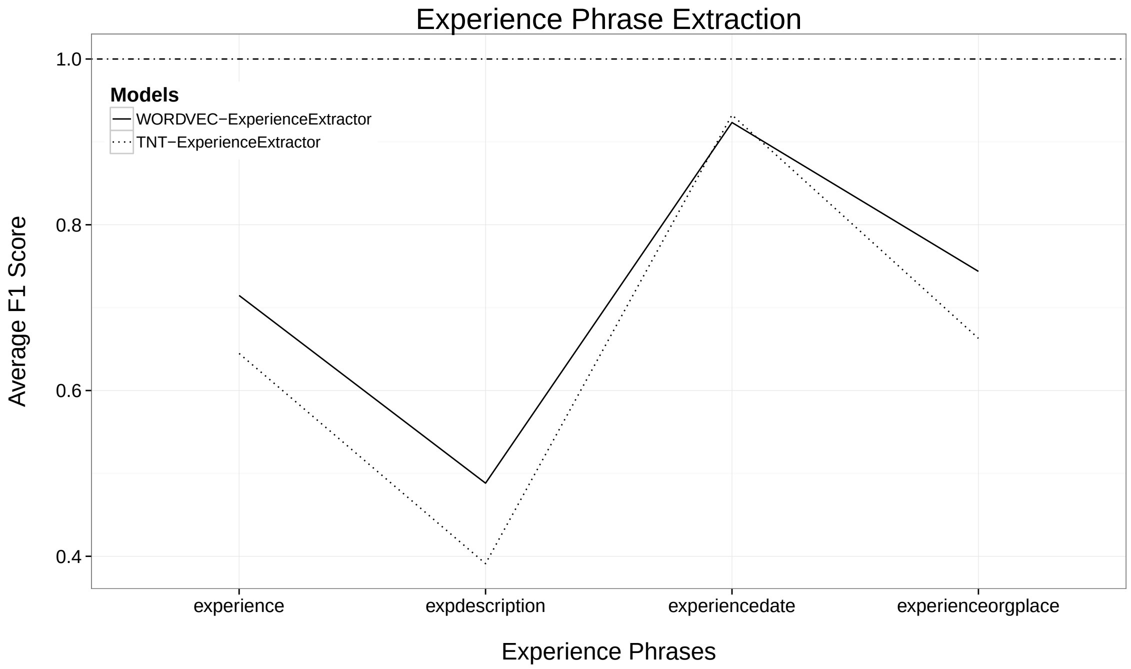


Figure 2.6: Results obtained by the TNT baseline model and the current candidate model.

dataset to train on, the accuracy of the CRF model decreases by only ~3% if we train on (a randomly sampled) 40% of the original dataset. This observation allowed for the exclusion of an outdated batch of CVs (roughly 30% of the original dataset) from the training partition without a decrease in performance.

Finally, we implemented new features for months, legal entities (like *GmbH*) and common job titles. We generated gazetteers for job titles and company names from external and internal collection of vacancies and replaced the latter feature by integrating the extended lists instead. Again, we report a higher performance for the CRF model than for the current baseline model. The results are illustrated in Figure 2.6.

2.4.2 Portuguese

To confirm the generalisability of the CRF model, we subsequently ported the CRF model to Portuguese. Since Textkernel is currently exploring the Brazilian market, the CV parsing model for Portuguese is being investigated by the rest of the team, providing an ideal opportunity to make use of the insights we gained so far.

In brief, we were indeed able to reproduce the results we reported for German. We generated word vector representations from the Portuguese Wikipedia as well as a new batch of sample CVs. Trained on an equally new partition, the CRF model outperforms the TnT baseline on both the personal and the experience section for all the relevant target fields. Combining the data sources for the vector generation does not improve the results any further.

Summary, Conclusion and Future Work

During the internship, different approaches to multilingual CV parsing have been examined.

We developed a new model that relies on the conditional random fields framework. We optimised parameters for representing each token in a high-dimensional vector space (data source, amount of data, and vector size) as well as training parameters for the *CRFsuite* (learning algorithm) and the actual training of the model (amount of data and features).

Compared to the HMM-based baseline model, the results of the CRF-based model combined with distributional semantic vector representations are promising throughout languages and different sections of target documents.

The main findings can be summarised as follows:

1. The CRF model clearly outperforms the baseline model.
2. Overall well-performing word vector representations are generated by:
    (a) Using a large sample of CVs, if available;
    (b) Combining all available data sources, otherwise.
3. Training the CRF model on vectors generated from Wikipedia data works equally well as using CV data.
4. Compared to the baseline model, the CRF model works much better for a much smaller set of training documents.
5. Integrating distributional representations of words highly improves the performance of the CRF model.

Changing the gold annotations to correspond to the current state of the annotation guidelines eventually did not improve the overall performance of the CRF model. One could argue in favour of the TnT results not being affected by the modified annotations at all. However, the sensitivity of the CRF model together with the fact that it works well for much less (annotated) training data allows for the claim that it might be possible to gain even better results by training on entirely flawless annotations. If for a particular language large data sets are not available, it thus seems feasible to annotate a much smaller set of documents and still obtain high accuracy results for the CV parsing task.

The above statements allow for many directions future work might pursue with regard to the study of conditional random fields using distributional semantic vectors.

To begin with, the issue of low recall is still not entirely solved. It could be examined whether the gazetteers (especially for job titles) could be improved by applying further cleaning, or if introducing line-based features has any effect.

Continuing along the same lines, the CRF model is currently working on single tokens only. Integrating compositionality for entire phrases could enable the system to recognise patterns in multiword expressions and complex compositional structures as well.

From a research perspective, it might be worth investigating how the vector size is influencing the representations in the vector space. Although we did not yield any improvements of the model by setting the number of dimensions very high (300 and above), there might still exist interesting properties of the corresponding representations that could be helpful for solving more fine-grained tasks, for instance.

Another feature that could be integrated in the CRF model is the POS-tag of the current word. Moreover, we are trying to bias the weights of the CRF to prevent the model from guessing the most frequent class of a certain phrase whenever it lacks a sufficient amount of feature information. An ensemble method combining the TnT model with the CRF approach might lead to further improvements as well.

We are planning on writing up a research paper on the contribution of using word vector representations instead of word entities as input for a CRF model for sequence labelling and strongly encourage you to refer to the paper for further technical details.

Personal and Professional Experience

I have had the wonderful opportunity to spend the summer working within a throughout inspiring research environment.

From the very beginning, Textkernel surprised me with the innovative and unconventional way the company operates:

Upon arrival, an additional work station had already been set up. It would later be directly connected to one of the 40–core computer clusters almost entirely dedicated to my research.

While the *Textractor* team is organising itself by the means of *Scrum*, an iterative and incremental framework for agile software development, the team spirit is clearly characterised by the ambition and open-mindedness of its members. The group's schedule includes *Tech Talks*, tutorials and reading groups on a regular basis.

The company culture at large is as informal as it is innovative. Textkernel provides free lunches, chair massages and overall good equipment to all its employees. Ping pong, outdoor lunches, and a lively Friday drinks tradition top off the usual 8–hour day that thankfully does not usually start before 09:30 AM.

For the entire period of the project, Textkernel paid a competitive salary.

Additionally, I was invited to join everyone for the *Textkernel Innovation Week* – a week in which each employee gets the chance to propose their own innovative idea and gather around it a team of colleagues to realise it – that found its end with a company trip to Rotterdam.

During the internship, I genuinely enjoyed the novel experience of contributing towards the products of a contemporary NLP company.

Using Git and Subversion, I quickly became more acquainted with current software versioning and revision control systems. Furthermore, I was familiarised with Perl as both the coding weapon of choice and powerful command line tool.

In contrast to predominantly theoretical university studies, the possibility of actually applying state-of-the-art techniques on large data sets has certainly been a valuable experience that highly benefited to my personal and professional development.

Textkernel researchers regularly attend academic conferences to remain at the cutting edge of innovation and provide accurate, fast and reliable understanding of documents.

For the last week of the internship, I received generous funding to participate in Coling 2014, the 25th International Conference on Computational Linguistics, which was held in Dublin and a great finish of my brief excursion to professional life.

References

[1] Thorsten Brants. TnT: A Statistical Part-of-speech Tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[2] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.

[4] Jorge Nocedal. Updating Quasi–Newton Matrices with Limited Storage. *Mathematics of Computation*, 1980.

[5] Naoaki Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs), 2007.

[6] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[7] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 807–814, New York, NY, USA, 2007. ACM.

[8] Torsten Zesch. *Study of Semantic Relatedness of Words Using Collaboratively Constructed Semantic Resources*. PhD thesis, Department of Computer Science, Technische Universität Darmstadt, 2009.

Appendix

A.1 Sections with Extracted Fields

Section	Extracted field	Equivalent
Personal	name	Full name
	address	Full address
	birthday	Date of birth
	birthplace	Place of birth
	phonehome	Home phone number
	phonework	Work phone number
	fax	Fax number
	email	Email address
	marstat	Marriage status
	nationality	Nationality
	gender	Gender
Experience	experience	Job title
	expdescription	Job description
	experiencedate	Period of time
	experienceorgplace	Company and location

Table A.1: Overview of selected sections and extracted fields