MACHINE TRANSLATION FOR NON-ENGLISH NAMED ENTITY RECOGNITION

by

Sara Mahboubeh Saghaei

B.Sc. (IT Engineering), Amirkabir Univerity of Technology, 2009

a Project submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the School
of
Computing Science, Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2012

APPROVAL

Name: Sara Mahboubeh Saghaei

Degree: Master of Science

Title of Project: Machine Translation for Non-English Named Entity Recognition

Examining Committee: Mr. Ghassan Hamarneh
Chair

Dr. Oliver Schulte, Associate Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Veronica Dahl, Professor, Computing Science
Simon Fraser University
Supervisor

Dr. Anoop Sarkar
Associate Professor, Computing Science
Simon Fraser University
Examiner

Date Approved: August 15, 2012

Partial Copyright Licence

Abstract

Parallel corpora, Often exploited for Machine Translation, have recently been used for monolingual purposes. Borrowing annotation from resource rich languages into resource-scarce languages is a technique known as Annotation Projection [26] that uses parallel corpora and word alignment to transfer annotations; It has been introduced as an alternative to the tedious and time-consuming task of building hand-annotated corpora for new languages. This technique is especially effective for semantic annotations such as Named Entity, since they are less affected by translation.

In this work we test the applicability of annotation projection to NER through two paradigms: One focusing on generating new German data and annotating it using English annotated data and another that focuses on adding new annotations to already existing German text and using them as training features.

We accompany machine translation with annotation projection which not only removes the restriction to parallel corpora and expands the methodology but also allows the use of monolingual hand-annotated corpora, relieving the bottleneck of English-side annotations quality.

We develop four training corpora by applying the two paradigms on two different corpora: parallel and singular. We train an NER model on each corpus for evaluation and compare the model quality with a baseline. The results show that the projected annotations can be noisy and inconsistent. Therefore, using them as target annotations reduces corpus and model quality; Whereas, as features alongside the original annotations they significantly improve the quality.

To my parents for their unconditional existence
To languages of the world for their undying curiosity

"There are four tongues worthy of the world's use:
Spanish to God, Italian to women, French to men, and German to my horse."

— The Holy Roman Emperor, Charles V.

Acknowledgments

I would first like to thank my Senior Supervisor, Dr. Oliver Schulte, for his great support and encouragement, and my Superviser, Veronica Dahl, for her financial support through NSERC Discovery grant 31611024.
And next, the people at the UKP research group at TU Darmstadt, under the supervision of Dr. Iryna Gurevych, who inspired the initiation of this project and taught me the basics.

My sincere gratitudes to my great friends, Iman Hajirasouliha, Bamdad Hosseini, and Amir Aghdam, who have lived this phase of my life with me.

My unconditional love to my mother for her unbroken affection and warmth,
And my utmost admirations to my father for being my primary source of inspiration all along.

My special thanks to the good old residents of 1357 and the squeaky chairs of Calhoun's Bakery who have kept me and my spirit up during the sleepless nights of thesis writing.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Named Entity Recognition

"Named entities are phrases that [most commonly] contain the names of persons, organizations and locations." [25]. For example, in the sentence, "U.N. official Ekeus heads for Baghdad", *U.N.* is an organization, *Ekeus* is a Person and *Baghdad* is a Location. Each one of them is an instance of a named entity.

The task of Named Entity Recognition is the automatic detection of such entities within a text. The categorization of named entities can be arbitrary and include many different types. It usually varies according to the purpose of recognition; the most common types are the name of a person, organization, and location [21].

Named entity recognition is an important and well-established task in information extraction systems [20]. It plays a fundamental role in a variety of natural language processing applications. In machine learning based modeling of monolingual NLP tasks, Named Entity annotations quite often appear along with other primary annotations, namely Parts-of-Speech and Chunk tags.

Named Entity Recognition can also be an end to itself; one of the most sensitive applications of it, with a high demand, is *redacting*, that is, removing privacy information, such as a person's name or address, from texts that are to be made public. Particularly, in the medical field, "De-identification" is the name given to the practice of anonymizing hospital records by removing patients information before making them available to researchers. [5]

Due to the relative ease of (specific-purpose and language-dependent) NER and its high

marketability, there has been a lot of work on it since the early 90s. Like any other monolingual processing tasks of NLP (Natural Language Processing), English has received the most attention and resources. Particularly, the growth of the field has been facilitated by a few prominent conferences. The first ones were the 6h and 7th Machine Understanding Conferences (MUC) in 1995 and 1998 respectively [9].
Between 1995 and 2002 a number of studies started to investigate NER in a multilingual fashion, comparing and contrasting scores on different languages, mainly on some European and a few Asian languages [25]. By 2002, the consensus among NER researchers was that the core parts of the NER task are in common among most languages. CoNLL 2002 and 2003 [25] followed this judgement and developed shared tasks and corpora of four European languages, namely English, German, Dutch, and Spanish, with the goal of encouraging a language-independent approach to Named Entity annotation. Section 2.3 discusses the CoNLL conferences in greater depth.

1.1.1 Challenges of NER

Although using lexical clues and word lists guarantees a certain level of accuracy for most applications of NER, building a general-purpose language-independent NER system with a high accuracy is not a simple task. Variability of the domain and language especially complicate it.
Beyond recognizing entities, an NER also has to get the class of the entity correctly. More often than not, the distinction of entity classes only happens through taking the context into account. For example, "is *Washington* a person or a location?" (from [3]), or is *Rogers* a person or an organization? They can be both, depending on the context. The following sample sentence from [3] depicts an array of ambiguities that can arise in NE detection:

> "Italy's business world was rocked by the announcement last Thursday that Mr. Verdi would leave his job as vice-president of Music Masters of Milan, Inc. to become operations director of Arthur Andersen."

Issues:

- Italy: is at the beginning of a sentence, so capitalization information is useless.
- The "'s" is not part of the name "Italy".
- The date is "last Thursday" rather than "Thursday".

- "Milan" is tagged as a part of an organization name rather than as a location.
- "Arthur Andersen" is an organization, not a person.

It is common to use a machine learning approach to NER, as well as to similar labeling tasks, known as *Sequence Labeling*. Sequence labeling is the problem of assigning each of the elements in a stream of tokens a categorical label. Commonly, labeling of the sequence is carried out through a joint segmentation of all the tokens in the stream; that is, the algorithmic determination of the optimal labeling for the entire sequence, rather than individual assignment of labels to tokens. This type of joint assignment is a means to provide the contextual hints necessary to disambiguate the categorization of NEs. Common models for sequence labelling are Hidden Markov Models or HMMs, Maximum Entropy Markov Models (MEMMs), and Conditional Random Fields (CRFs). These three models are briefly described in Section 2.1.1.

1.2 Machine Translation

Machine Translation (MT) is using computers to automatically translate from one natural language into another. It is one of the key focuses in the field of natural language processing. Various Artificial Intelligence based approaches to this problem have been attempted since the 1950s, until two decades ago when the machine learning approach, known as Statistical Machine Translation (SMT), started to take over. SMT uses large parallel corpora and machine learning algorithms to model translation. A parallel corpus (also known as parallel text, bitext, or multitext) is a collection of sentences in two different languages, with each sentence aligned to its corresponding translated sentence in the other language.
Using SMT, building an automatic translation machine from any language to any other language is a matter of having the right amount of parallel text and a few weeks of training time [4]. SMT has led to the rapid progress of MT – both the commercial MT and the academic state-of-art–over the past two decades.
Quality of the translations produced this way depends on (1), the "quantity, quality, and the domain of the training data" [19], (2) the amount of linguistic knowledge that is automatically acquired from the corpus, or manually provided [14]. Named Entity annotations too, as a linguistic knowledge, are often used for improving translation quality in training of SMT systems [3] (however, the reverse direction has been rarely explored).

SMT has grown dramatically over the past few decades and the translation quality has improved a lot. The question of whether and to what degree MT can replace human translation/translators have been raised. But it has also been argued that the usefulness of MT is not solely reliant on a human comparable translation quality. Even at a low quality MT is useful, given the appropriate application for it [4]. At the basic level, MT serves as a context-driven word-for-word translation which according to [13] addresses most of the users wish, since they can "generally recover from scrambled syntax".
In Section 2.2, I will talk about Moses, a freely available implementation of SMT at its current state, that facilitates building an automatic translator from scratch. It is a widely popular toolkit within the SMT research community.

1.3 Motivation

Annotated corpora are a crucial resource for most language processing tasks, including Named Entity Recognition. They serve the double purpose of providing data for machine learning modeling and the reference for the evaluation of such a model's quality. Building these annotated corpora takes a lot of human effort and time. Plus, the majority of these efforts address the advancement of English tasks. Resources for other languages are even rarer.
To make matters worse for Named Entity tagging, annotation schema can vary among corpora. As mentioned before NER corpora come with a variety of tagsets. Linguistic annotation of a corpus is not a straightforward task to begin with; trying to conform to a uniform schema makes it even harder.
All in all, obtaining an annotated corpus remains the single most crucial bottleneck in any language processing task with a machine learning approach, including NER.
The main purposes of this project, therefore, is firstly, to take an alternative approach to filling up this shortage of resources, specifically for non-English languages. The basis of these alternative approaches is on *recycling* currently available resources, using advanced tools of the field, that is MT and English NER, in order to adapt them to the special needs of a particular task. More specifically, the goal is to introduce and evaluate a methodology based on automatic translation to quickly develop new annotated corpora or to compensate for their shortage in languages other than English.
The NER modeling tool that we use in this project is Stanfords NER engine, which is one

of the state-the-art language independent engines. It has performed well in CoNLL shared tasks and has continued to evolve since then [8]. Based on the performance metrics of CoNLL and Stanford the German NER models lag behind the English ones by 10-20% [1]. Spanish and Dutch models perform worse than English too, but not worse than German. The main reason for this performance gap must be the bigger size of English training data [25]. This lower performance in non-English languages is one of the primary motivations of this project.

Secondly, the issue of language independence in NER, that was raised by CoNLL 2002 and 2003, is addressed. CoNLL promoted independence from language in NER engines and in this way extended the attention to non-English languages. In this project, however, we aim to take an opposite approach to the same problem: rather than independence, we use language-specific models to reinforce one another.

The key idea that bridges the connection of NER models is the observation that as the words of one language translate into another, named entities remain named entities, and of the same type too. For instance, the word "United States" in English, which is a location NE, once translated to French would change into "Etats-Unis" but still be an NE to a French NE annotator. Sometimes even the lexicon of the entity remains unchanged, such as translating persons names. In other words, *translation preserves the property of being a named entity*, because this is a semantic feature, whereas translation might change syntactic annotations like parts-of-speech and chunk.

The seeming straightforward correlation of named entities across languages and the strong need for more resources for non-English languages, led us to design a number of experiments to test the degree to which this observation holds and can contribute to the quality of NER models in a non-English language.

I have focused on German for the sake of the experiments, but there is no reason they should be limited to it. We simply picked one of the CoNLL languages since resources needed for experiments were available for it.

[1] http://nlp.stanford.edu/projects/project-ner.shtml

1.4 Project Description

In this project, we use translation as a means to develop annotated corpus. The idea is to project annotations to unannotated corpus from its sentence aligned translation text over word alignment. This method is known as *Annotation Projection* [26]. We use English text and its annotations as the source of projection and German as the target.
Translation and annotation projection collaborate within two different settings in this project. The first setting, or *Paradigm A*, is focused on generating new German text and annotating it. The second setting, *Paradigm B*, concerns with adding new annotations to already existing German text and using them as training features:

- In Paradigm A, English text is 1) NE-annotated, 2) translated into German, 3) its annotations are projected to the German translation to build a new training corpus.
- In Paradigm B, annotated German text is 1) translated into English, 2) its English translation is NE-annotated, 3) the English annotations are projected back to the German text to make a training corpus with two NE-annotations.

Paradigm A works on the translated texts and projected annotations directly. It is quite likely that both of these processes introduce a lot of noise into the training data. The second paradigm is therefore, designed for the sake of a more sophisticated approach to exploiting both techniques: It uses translation only to perform annotation projection and uses the projected annotation only as features, rather than training targets.

Translation is another design factor: Both Paradigms are applied once to (NE-annotated) corpora without parallel text and once to parallel corpora (without NE annotations). For annotated corpora we use CoNLL 2003 datasets and for parallel corpora we use EuroParl. (See Sections 2.3 and 2.4 for information on datasets). Translation and annotation are at trade-off with it each other: annotated corpora does not have a parallel text and parallel text does not have annotations.

Alternating the two design factors of paradigm and translation brings us to four different ways of building training data and training a new NER model. Table 1.1 shows an overview of the properties of these models and their paradigms.

The highest ambition of this project is to establish a fully automatic methodology that removes all resource bottlenecks: automatic translation, automatic Named Entity tagging

Table 1.1: Our four NER models and their properties

Name	Corpus	Translation	Annotation	Project Annot. As
CoNLL-A	CoNLL2003	MT:En→De	available	Target
CoNLL-B	CoNLL2003	MT:De→En	available	Feature
EuroParl-A	EuroParl	available	Automatic	Target
EuroParl-B	EuroParl	available	Automatic	Feature

on the English side and automatic back projection of the annotations over automatically obtained word alignments. The success of this methodology would serve as evidence to maturity of Machine Translation and would prove the possibility of its contribution to monolingual NLP tasks.

The rest of this document is structured in the following way: The next section is an overview of related work. The next chapter introduces the Tools and Datasets employed in this project. Its subsequent chapter, Chapter 3, contains a more detailed description of the models and the paradigms. In Chapter 4, Evaluation, the experiments, their results, and a discussion of the result are provided. At the end the conclusions and future works are reviewed.

1.5 Related Work

Parallel corpora, used for Machine Translation tasks, have very recently been found useful for monolingual purposes as well. Borrowing resources from resourceful languages has been introduced as an alternative to the tedious and time-consuimg task of building hand-annotated corpora. A number of works during the past decade have adopted the annotation projection paradigm to this end. The paradigm, as mentioned previously, is based on transferring annotations from resource-rich languages (English mostly) to resource scarce languages.

One pioneer work in using parallel corpora to this end is the work of Yarowsky et al [26] from 2001. They introduce an annotation projection framework over multilingual corpora with the purpose of monolingual modeling of four different types of linguistic annotations, including Named Entities. Their work has turned into a starting point for works under this topic.

Bentivogli et al [2] in 2004 developed an Italian Word Sense Disambiguation (WSD) annotated corpus from the annotations that were projected from parallel English corpus. Another

experiment involving WSD annotations in 2002 [12] aligned the words of George Orwell's novel, *1984*, across six languages and derived as many senses as possible for a set of 33 English nouns in those languages. They showed that these fully automatic results were comparable with human WSD annotations.

Pado et al [23] introduced a sophisticated framework based on annotation projection to extend semantic role annotation from English to new languages. They use their framework on German-English corpora with good results.

The work of Ehrmann et al [6] in 2011 is one of the few works that solely focused on Named Entities. They work on developing a large multilingual NE-annotated corpora through projection from English for generic uses.

Beyond Semantic annotations, Rebecca Hwa et al, [10] and [11], have proved the prospect of annotation projection for syntactic annotations. In one of their works they project POS tags and dependency trees from English to Chinese and develop a new Chinese parser based on it [10].

The two main bottlenecks for output quality that are commonly recognized in all annotation projection works are: 1) English-side (or source-side) annotations and 2) word alignment quality.

Beyond the basics of the paradigm, –the type of linguistic annotation and the choice of languages–, studies on this topic vary a great deal on their design factors. Below is a summary of these variations which at the same time aims to clarify the factors involved in the task of projection and its context of employment.

- *Word alignment.* Annotation Projection always uses word alignment as the connecting bridge. Most works, like ours, use automatic word-alignment tools for obtaining the alignments. But some others, like [6], attempt at alternative methods centring on string matching techniques.

- *Using the projected annotations*: Except for Yarowsky et al [26] who project many annotations at the same time and use the non-primary ones as training features, as well as us using the primary projected annotation as a feature, almost always the projected annotations take the place of the target annotation in further modelings, if any is done.

- *Objective.* A number of studies use annotation projection to build new corpora for public use: Ehrmann et al [6] and Bentivogli et al [2] are the examples. Mostly others,

including our work, and the works of Yarowsky et al and R Hwa et al mentioned above, use them for the purpose of building better monolingual models.

- *Evaluation.* Evaluation is one of the trickiest parts of this task, given the scarcity of hand-annotated resources that can be used as references –which itself is the very motive behind taking this alternative approach. Manually evaluating the outputs also require the knowledge of the foreign language. Many of the studies that focus on developing a generic annotated corpora have nevertheless, taken all the measures in manually evaluating their works(including [2] and [23]). Others however, have turned to modelling to determine the quality of the output corpus. We take the second path as well.

This work has a number of substantial innovations compared to previous works in this topic:

1. Accompanying machine translation with annotation projection to remove the restriction to parallel corpora and relieve the bottleneck of English-side annotations by using monolingual hand-annotated data is a novel approach that has not been taken before.

2. Indirect use of the primary projected annotation, i.e., as training feature rather than as the target, with the aim of reducing the noise of projection is also unprecedented.

3. Lastly, we use automatically obtained annotations together with manual annotations in one corpus. This facilitates evaluation in an automated and accurate way which has not been attempted before.

Chapter 2

Tools and Datasets

2.1 Stanford Named Entity Recognizer

Stanford NLP group provides a free java implementation of a Named Entity Recognizer [8]. The engine uses CRF (Conditional Random Field) method of Sequence Labelling and a feature extractor specifically written for the Named Entity Recognition task. The software package is ready to download at their website, together with a number of pre-compiled models.
In the next section, Sequence Labelling methods, including CRF, are briefly explained. Next, an overview of the Stanford models feature sets are given, followed by an overview of their models. Finally, the procedure of training a new model with Stanford engine is described.

2.1.1 Sequence Labelling

Conditional Random Fields is a framework of probabilistically modeling segmentation and labeling of sequence data [17]. The need for labeling sequence data arises in various scientific fields. In computational linguistics, its applications include but are not limited to topic segmentation, part-of-speech and other annotations (including named entity) tagging, syntactic disambiguation, and information retrieval.
Traditionally, Hidden Markov Models and stochastic models have been used to tackle this type of problem. But more efficient methods have developed over the past few decades.

Hidden Markov Models (HMM) are *generative* models; meaning that, they need to "enumerate all possible observation sequences" in order to calculate the likelihood parameters of their models. An *Observation*, for instance in text labeling, is an input word token. HMM models calculate joint probabilities of observations and label sequences.
A *conditional* model, on the other hand, estimates the probabilities of label sequences based on a given observation sequence, instead of generating all possible observation sequences. This makes sense, beside being obviously more efficient, since the observation sequence is fixed at run time anyway.
Maximum Entropy Markov models, therefore, as conditional models maintain this advantage over HMMs. At each source state, which is a label in a sequence, an input observation maps to a distribution over the next possible states (labels). Experiments show the increase of recall and precision in MEMMs over HMMs. [17]
However, MEMMs do have a disadvantage known as the "label bias" problem. MEMM Labeling can favor those state-to-state transitions with fewer outgoing transitions, since the "mass" transferred at each transition is only normalized at state level, and does not consider the other transitions in the model. CRF takes on this issue and improves it: Instead of a per-state model for probabilities of the next state, it maintains a single model for the joint probability of the entire labels sequence, given the observation sequence. CRF can also be thought of as a "finite state model with unnormalized transition probabilities." [17].

2.1.2 Stanford Feature Set

The second part of the Stanford NER engine is a large feature factory class, with a lot of features and an easy interface for extracting new features. Following is a list of more important features they include in their trainings:

- Word features: current word, previous word,
- Surrounding words: Next word, all words within a window
- Orthographic features. Example: Sara → Xxxx, MUC-6 → XXX-♯, etc.
- Prefixes and Suffixes. Example: Sara <S, <Sa, <Sar, and ara>, ra>, a>.
- Label sequences
- Lots of feature conjunctions

- and Distributional Similarity features

Distributional similarity. Distributional Similarity is a model generalization method for tackling the issue of data scarcity. Distributional Similarity (or DistSim) functions measure similarity of words in order to estimate the probabilities of previously unseen data. Similarity of words is determined through various functions based on the similarity of their distribution over contexts in which they appear.
Traditionally, models would equate the conditional probability of an unseen component, with the probability of the seen part. For instance, P(is_NE|Bratislava), if the word Bratislava has not occurred in the training data, would be equal to P(is_NE). Through the Distributional Similarity method however, similarity-based clusters over the words of language (for which a very large lexicon corpus is required) is obtained and are assigned to the words. The probability of unseen cases then, are estimated based on probabilities of the words in the same cluster and their degree of similarity to the unseen word [18]. Distributional Similarity features boost model performance, but increase its size and running time.

2.1.3 Training a New Model with Stanford NER

To train a new model with Stanford NER, first the training features should be determined. A lot of the features are implemented in the engine and can be enabled in training by including their names among the features list. It is possible however, to implement new features as well through two main classes: CRFClassifier and NERFeatureFactory. The latter deals with the extraction of a feature from the input text, and the former with how to use the feature in training.
Once the features are set, it is time to prepare the prop file. Prop file (a text file with .prop suffix) is a configuration file that is input to the classifier. All the input, output files, the features and any other training setting, e.g., optimization, are specified in this file.
In this project, the new models that we train did not need any new features that were not already implemented. The baseline feature list was borrowed from the prop files of Stanford German models with minor changes. In Paradigm A models the prop file does not change over the course of different experiments, only the dataset does. In Paradigm B, we only add the projected annotation as a new generic feature to the prop file (More on this in Chapter 3). More information on training a new model with Stanford NER can be found on their

website[1].

2.1.4 Stanford Models

Three pre-compiled English NER models are provided in the Stanford NER package. One is a 4-class model trained on CoNLL 2003 data. Another is a 7-class model trained on MUC data. And the third one is a model trained on CoNLL, MUC, and ACE[2] corpora with only 3 entity classes –intersecting the entity classes of all three. In this project, we use the Stanford CoNLL model whose tagset conforms to CoNLL 2003 annotation: Person (PER), Location (LOC), Organization (ORG), and a fourth catch-all class of Miscellaneous (MISC) (See more on CoNLL data in Section 2.3).

Two German models are also provided with the Stanford package, which are developed by Faruqui et al [7] using the Stanford engine. They are both trained on CoNLL 2003 German Corpus but generalized with different Distributional Similarity lexica. One of them, *deWac-generalized*, is generalized with a large distributional similarity lexicon formed on the deWac corpus [1] and the second one, the *HGC-generalized*, is generalized with HGC (Huge German Corpus). DeWac corpus is a big corpus scraped off ".de" web data ranging over a wide genre of content; therefore, deWac-generalized model is useful for all kinds of documents. HGC is a more homogeneous and clean corpus of news-wires; So the HGC-generalized performs better on this specific genre, but not necessarily on others[3].

2.2 Moses

Moses is a free open source implementation of SMT. By providing a training corpus of parallel text in the two languages of interest (translated texts aligned at the sentence level), Moses will train an automatic translator between the two languages, in the chosen direction. Moses supports three types of SMT known as Phrase Based SMT, Syntax-based or Hierarchical SMT and a third type which is an extension of the first with the possibility of adding extra linguistic information, Factored SMT. The basis of training is on co-occurrences of

[1] http://nlp.stanford.edu/software/CRF-NER.shtml

[2] A corpus of broadcast news, broadcast conversation, newsgroups, weblogs data, annotated for Automatic Content Extraction (ACE) technology evaluation program in 2005, with 5 entity classes: Person, Organization, Location, Facility, and Geo-Political Entity. For more info, see http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2006T06

[3] See http://www.nlpado.de/ sebastian/software/ner/README.1.1.1 for more information.

continuous sequences of words in the aligned texts. The three types of SMT are mainly different in how they segment sequences[4].
Translation in Moses is a two-step process, as it is generally in SMT, with two different components:

1. the training pipeline, which takes the raw data and builds a model for translation.
2. the decoder, which takes the trained model and an input sentence, and translates it into the target language.

The training pipeline, in turn, consists of a word-alignment component, typically GIZA++ (See below), to match words in parallel sentences, a language model, a statistical model of the target language that helps choose the right sentence in the target language from among the options the translation model provides, and tuning to optimize translation weights based on the training/development corpus.
The decoder's job is to find, based on the translation model obtained from the training pipeline, the most likely translation in the target language (or an n-best list of translations) for a given sentence in the source language. Moses implements methods to optimize the performance of this potentially enormous search problem. Technical details of these components can be found in the original paper [16].

Before training with Moses, data has to be prepared. Typically, the training data is tokenized and lowercased before training, demanding the test set to be prepared the same way as well. The output of testing is then re-cased (capitalization turned back to normal) and de-tokenized. Changing the capitalization of training data is mainly due to data scarcity problem and sometimes to overcome the poor capitalization in input data (which happens if the data contains informal and unprocessed content).

2.2.1 GIZA++

GIZA++ [22] is an extension of the original GIZA package, which was developed as part of the EGYPT toolkit, at Language and Speech Processing laboratory of Johns Hopkins University.
Word alignment is the most time consuming part of SMT training. GIZA++ is a freely available package for this purpose, which implements the IBM Models of alignment [22].

[4]More on the three types of training at http://www.statmt.org/moses/?n=Moses.Tutorial

Table 2.1: CoNLL Data Format ('O' means part of no chunk, or no named entity)

Word	POS	Chunk	NE
U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

GIZA++ aligns words in a one-to-many fashion. If run from German to English, for instance, each German word would at most align with one English word, but English words might have more than one correspondent. Therefore, sometimes, as in Moses too, the aligner runs once in each direction and their intersection is obtained for higher accuracy.

2.3 CoNLL Dataset

CoNLL 2002 and 2003 were shared task conferences centred around language independence in Named Entity Recognition. A multilingual annotated corpora were provided through these shared tasks which are a collection of news articles. CoNLL 2002 addressed two languages: Spanish and Dutch. 12 different learning systems were applied to these two languages. CoNLL 2003, focused on English and German: 16 learning systems were now competing. The Stanford NER, with a slightly weaker system back then compared to now, stood in the top third in both English and German in the 2003 task. especially, in the German task there was very little difference between the top three systems.

CoNLL Data Format. CoNLL presents data in a columned format, with the text token appearing on the first column followed by various annotation information. In the English data, the annotations are POS, syntactic chunk, and NE. the PoS tags follow the Penn Treebank PoS convention and the chunk tags follow the CoNLL 2000 shared task convention [24], which contains eleven different categories. Table 2.1 shows an example of CoNLL data format.

2.4 EuroParl Dataset

EUROPARL [15] is a freely available dataset gathered from the proceedings of European Parliament in 21 languages. Alignment at sentence level with the English version is done for all languages and parallel texts are made available. The size of these parallel texts vary for different languages, but it is in the general range of 400K to about 2M sentences. The German-English corpus contains 1.9M sentences with 44M and 47M words in German and in English respectively.
The datasets are available for download at the EuroParl website[5].

[5] http://www.statmt.org/europarl/

Chapter 3

Models and Methods

3.1 Methods

We use annotation projection within two different paradigms: paradigm A and B. We apply both paradigms once to CoNLL data and its machine translated parallel text and once to parallel texts of EuroParl. The paradigms we carried out are explained in abstract terms here, since the details vary between models; but specific the model descriptions later into the chapter will clarify the details of the paradigms as well.

Paradigm A. The corpus extension paradigm. In this paradigm, new unannotated German data is either obtained through MT or as part of the parallel corpora. Annotation is acquired for the new data through projection and the projected annotations are regarded as the target tag of training.

Paradigm B. The feature set extension paradigm. In this paradigm, an English translation for an existing German data is acquired. The English translation is automatically annotated and its annotations are projected over to the German data. The German data itself, is already manually or automatically annotated, leaving it with two sets of NE annotations: original and projected. The projected annotation is used as a feature and the original annotation as the target of training.

In both paradigms, at test time, same procedure is carried out for obtaining testsets.

The four models developed in this project differentiate based on 1) the paradigm they adopt and 2) the dataset they apply it to.

The models that work with CoNLL dataset include annotation on German side, but EuroParl models need to obtain it automatically.

EuroParl models come with manually translated parallel text, but CoNLL models need to obtain the translation texts.
Essentially, there is a trade-off between translation and annotation in experimenting with annotation projection. Translation and annotation are also the main performance bottlenecks.

3.2 Translation

We developed a Moses baseline phrase-based translation system from English to German and another from German to English as instructed on Moses website [1]. WMT10 [2] parallel corpora were used for training, which are a multilingual corpora of EuroParl and parallel News Commentary data.
The GIZA++ word alignment is obtained in the process of training that is later used for annotation projection.

3.3 Baselines

I train two baseline models: One on CoNLL German data, the other on EuroParl data.
The first baseline model is trained on German CoNLL data (CoNLL-baseline) using the same feature set as the deWac-generalized German model (See Section 2.1.4), except for the distributional similarity features.
The second baseline model is trained on EuroParl German data, using the same feature set. Obtaining the training set for this model requires automatically annotating the EuroParl German data, since it is not hand-annotated. DeWac-generalized was used for annotation, because of its wider domain coverage than HGC (which only covers news-wires).

3.4 Models

3.4.1 CoNLL Model A

This model is based on CoNLL corpora and our corpus extension paradigm (paradigm A), which involves automatic translation and using projected annotations as target tags.

[1] http://www.statmt.org/moses/?n=Moses.Baseline
[2] http://www.statmt.org/wmt10/

The training corpus for this model is obtained through the procedure depicted in Figure 3.1.

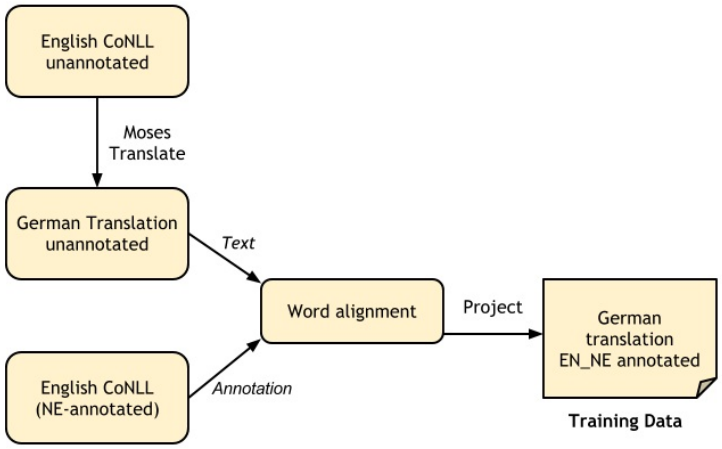

Figure 3.1: Procedure of obtaining Training Data for CoNLL-A Model

Translation

We stripped English CoNLL data off its annotations, put it into batch text format and lowercased it (it is already tokenized). We then translated it into German using Moses.
We then re-cased and reverted the German translation into one token per line CoNLL format, but with no annotations yet.

Projection

We wrote a small *projector* program in Python to perform the projection. It takes as input the original English data in CoNLL format, with two columns of text and NE, the German data in the same format but with only one column of text, and their alignment. The way the program works is that it reads in the entire alignment file first and builds a mapping of words per sentence. Second, it reads in the entire English text and builds a map of words to NE tags per sentence. Then reads the German text and matches each word with its aligned English word, from there to its NE tag, and outputs the word and its obtained tag on one line. The output is the German text with Named Entity tags projected from English corpus. A sample projection procedure is shown in Table 3.1.

Table 3.1: Projection Example

English		Alignment			German	Projection Result	
the	O	the	→	Die	Die	Die	O
European	I-ORG	European	→	Europäische	Europäische	Europäische	I-ORG
Union	I-ORG	Union	→	Union	Union	Union	I-ORG
was	O	was	→	hat	hat	hat	O
right	O	right	→	zu Recht	zu Recht	zu Recht	O O

Training

The appendix A shows the content of the .prop file used for training of the German translation text. The same feature set as the baseline models are used for it. Once the prop file and the input data are ready, running the training is a matter of running one command:

```
bash$ java -cp path-to-stanford-package/stanford-ner.jar \
edu.stanford.nlp.ie.crf.CRFClassifier -prop name-of-prop-file
```

3.4.2 EuroParl Model A

This model works with EuroParl corpora as described in Section 2.4 and paradigm A, corpus extension. The process of preparing the training data is visualized in Figure 3.2.

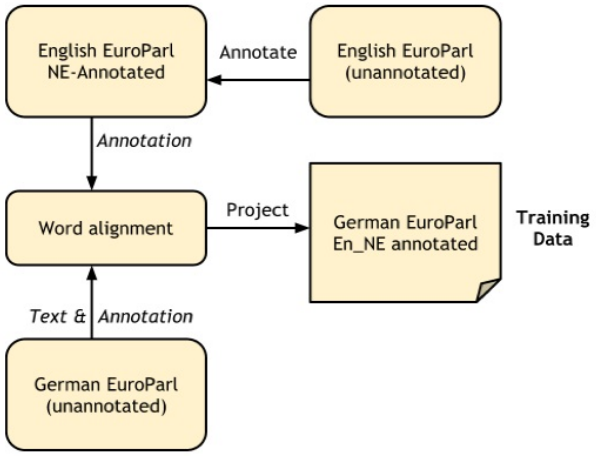

Figure 3.2: Procedure of obtaining Training Data for EuroParl-A Model

English EuroParl is annotated with Stanford CoNLL model. Projection and training are

done the same way as the previous model.

3.4.3 CoNLL Model B

This model is based on applying the B paradigm to CoNLL dataset. Figure 3.3 summarizes the process. The steps that we carry out are as follows:

1. The German CoNLL, stripped of its annotations, is first translated to English using Moses.
2. The English translation is NE tagged by Stanford CoNLL model.
3. The results are projected back into the original German text using word alignment and the projector script.

The new German text now has three columns: word, projected NE, German original NE. The training uses the original annotation as the target and the projected annotation as an independent feature. The details of translation, projection, and training are as explained before.

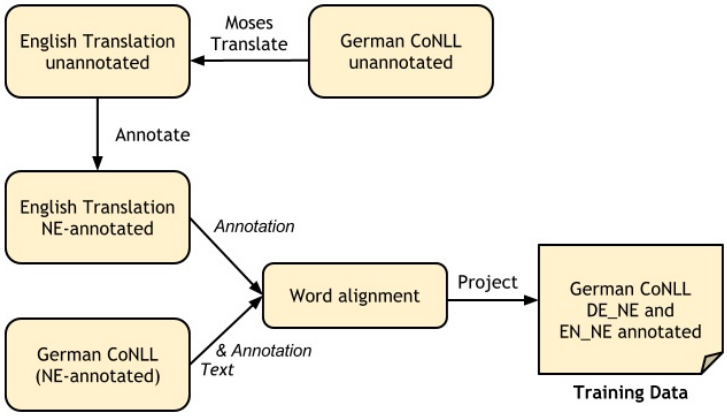

Figure 3.3: Procedure of obtaining Training Data for CoNLL-B Model

3.4.4 EuroParl Model B

This model works with paradigm B and EuroParl training dataset as described in Section 2.4. Figure 3.4 visualizes the procedure: The English and German EuroParl are independently

NE-annotated using Stanford CoNLL and Stanford deWac-generalize models, respectively. The projector code projects the English annotations over the word alignment and into the German dataset. The resulting training data uses its own annotation as the target tag and the projected annotation as an independent feature. The details of annotation, projection, and training are as before.

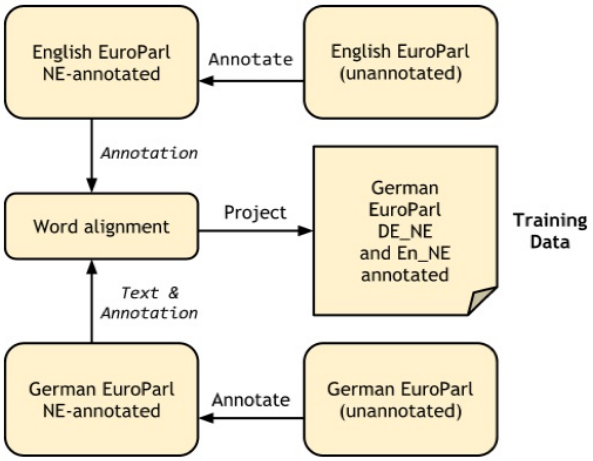

Figure 3.4: Procedure of obtaining Training Data for EuroParl-B Model

Chapter 4

Evaluation

4.1 Experiments

In this section I will discuss and analyze the results of Paradigm A models, followed by the results of Paradigm B models, followed by a discussion of all results at the end.

4.1.1 Paradigm "A" Experiments

The purpose of paradigm A experiments are evaluating the new annotated corpus.

CoNLL-A Evaluation

I compare the performance of CoNLL-A model with the CoNLL baseline model by running them both on CoNLL test-a dataset.
the command below can be used for running a model on a test file:

```
bash$ java -cp path-to-stanford-package/stanford-ner.jar \
edu.stanford.nlp.ie.crf.CRFClassifier -loadClassifier \
path-to-model -testFile path-to-test-file
```

The command will automatically generate the evaluation metrics, precision, recall, and F-measure, for each entity class, through comparing the model output with annotations on the file. Table 4.1 shows the result of running the baseline CoNLL model on the test set and table 4.2 shows the result of running CoNLL-A on the test set.
A big drop of about 10-20% in model quality is visible in the second table. The average

Table 4.1: Baseline CoNLL Model on CoNLL Test

Entity	Precision	Recall	F-measure	TP	FP	FN
Location	78.91	61.24	68.96	722	193	457
Misc	77.14	41.71	54.14	415	123	580
Organization	83.33	57.21	67.85	710	142	531
Person	74.38	47.50	57.98	665	229	735

Table 4.2: CoNLL-A Model on CoNLL Test

Entity	Precision	Recall	F-measure	TP	FP	FN
Location	65.52	40.46	50.03	477	251	702
Misc	73.60	33.07	45.63	329	118	666
Organization	58.39	32.23	41.54	400	285	841
Person	63.0	33.57	43.80	470	276	930

precision, recall, and F-measure of the baseline are 78.5% , 52.1% and 62.63% respectively, whereas the new model only gets a precision and recall of 61.6% , 31.8% , and 41.95% respectively.

Analysis. In order to find the causes behind this bad result, I ran the Stanford model separately on the translated training data. The result showed a precision of about 60% and a recall of 36% on average, revealing disagreements between the German models' output and the projected annotations on file. I looked into these annotations for the sources of disagreement. A sample of possible cases of disagreement in annotation are shown in table 4.3.

Annotation Error. The first example shows the word *deutsches*, meaning *German*, and the word, *britisches*, meaning *British*. The first one always gets tagged with the Misc label by the two German models, but the latter is never recognized as an NE. While in the English text, both British and German are tagged with Misc. Another similar example is that of *Veterinärausschuß*, meaning *veterinary committee*, which is rightly recognized as an organization by the german models, but 'veterinary committee' is not an NE in the English text. This is evidence of inconsistency of annotation between corpora especially if they are of different languages.

Another inconsistency is in whether titles of people should be tagged as part of a PERSON entity. One example is *Komissar Fischler*, (a repeated name in the corpus) which one of the

Stanford models fails to recognize it as a Person altogether, and the other one does so only about the "Fischler" part, and not "Komissar". In the original English text, "Kommissar", sometimes translated to Commissioner, sometimes to Inspector, is sometimes recognized as part of an entity and sometimes is not.

Translation Error. There are a few cases where an annotated word is actually translated wrong so that it is no longer a named entity. Examples are when *EU* (European Union) did not get translated at all; instead, its preceding determiner, *der*, was repeated, causing the aligner to align *EU* with *der*. As a result *der* has received an I-ORG tag.

Sometimes translation causes the title problem mentioned before. Translation into German sometimes adds titles to names of people (such as Herr (Mr.) or Kommissar (commissioner or inspector)) that do not exist in the original text. This causes the word alignment to extend the annotation of the name to the title too and the title would become an NE, while the German annotator will not recognize it as such and neither did the English annotation. Sometimes too, tokenization and capitalization issues which are the side effects of automatic translation cause similar problems. Oftentimes the names of people are not recognized by German annotators because they are written in lowercase.

An example of the tokenization problem is the phrase, "Ain't no telling", which is the name of a Jimi Hendrix song and a miscellaneous named entity. This phrase fails to translate properly into German because Ain't is tokenized into Ai-n't and is left untranslated. So the German annotators do not recognize it as an NE. However, there is no telling whether there is a proper German translation for the phrase which would be recognized by German models or should such phrases remain in their original language in order to be picked up by the foreign language annotators. These cases cannot be handled easily in an all-automatic setting. However, they are not very common.

Finally, Some of the cases are simply a case of mixed up target syntax by the translator that confuse the German models. the phrase *"die Kommission Chefsprecher der Nikolaus van der Pas"*, translated from *The Commissions spokesperson Nikolaus van der Pas*, is in wrong German format and it is annotated wrongly by both German models. □

A few observations should be made from these results:

- Manual annotation can be inconsistent with itself and/or with another manually annotated corpus, especially, if it is in another language, therefore, causing incompatibility

Table 4.3: Sample of differences in annotation between translated+projected data and German models annotations

Word	Projected	deWac	HGC
deutsches	I-MISC	I-MISC	I-MISC
britisches	I-MISC	O	O
Veterinärausschuß	O	I-ORG	I-ORG
Werner	I-PER	I-ORG	I-ORG
zwingmann	I-PER	O	O
EU→der	I-ORG	O	O
Kommissar	I-PER	O	O
Fischler	I-PER	O	I-PER
Ai	I-MISC	O	O
n't	I-MISC	O	O
no	I-MISC	O	O
telling	I-MISC	O	O
die	O	O	O
Kommission	I-ORG	I-ORG	I-ORG
Chefsprecher	O	I-ORG	I-ORG
der	O	I-ORG	I-ORG
Nikolaus	I-PER	I-ORG	I-ORG
van	I-PER	O	O
der	I-PER	O	O
Pas	I-PER	O	O

Table 4.4: CoNLL-B Model on CoNLL Test

Entity	P	R	F1	TP	FP	FN
I-LOC	79.72	73.71	76.60	869	221	310
I-MISC	81.79	55.98	66.47	557	124	438
I-ORG	82.56	61.80	70.69	767	162	474
I-PER	82.12	53.79	65.0	753	164	647

that hinders annotation projection.

- Capitalization and tokenization are frequent cases of confusion within the translated corpora.

At the end, the poor results of the experiment in this section prove that a more sophisticated approach need to be taken in order to effectively exploit this cross-lingual connection. Even if in principle there should be a direct connection between English and German NE annotations, in practice, it cannot be simply assumed. And that is the purpose of designing Paradigm B models.

4.1.2 Paradigm "B" Experiments

The purpose of paradigm B experiments are evaluating the models with an added feature from projection.

CoNLL-B Evaluation

In this experiment CoNLL-B model performance is compared with CoNLL baseline by running them on CoNLL test-a dataset. Table 4.1 contains the baseline results; Table 4.4 shows the result of the CoNLL-B model run. With an average precision, recall, and F-measure of 81.4% , 61% , and 69.74% the new CoNLL model stands above the baseline precision of 78.5% , recall of 52.1% and F-measure of 62.63%İ also run the models on their training sets where both of them did exactly the same (with close to 100% accuracy).
The only difference between these two models is the extra feature projected from English translation, so there is no doubt using projected annotation as features improves model quality.

Table 4.5: deWac Model on CoNLL Test

Entity	P	R	F1	TP	FP	FN
I-LOC	79.56	73.62	76.48	868	223	311
I-MISC	79.75	52.26	63.15	520	132	475
I-ORG	81.13	61.32	69.85	761	177	480
I-PER	89.65	69.29	78.16	970	112	430

Table 4.6: HGC Model on CoNLL Test

Entity	P	R	F1	TP	FP	FN
I-LOC	83.56	77.18	80.25	910	179	269
I-MISC	80.15	53.97	64.50	537	133	458
I-ORG	84.11	63.98	72.68	794	150	447
I-PER	92.45	77.86	84.53	1090	89	310

Stanford Models. Although there is still more to add to this model in order for it to compete with state-of-the-art (including distributional similarity features, as discussed before –see Section 2.1.2), I tried running Stanford models on the same test set to see exactly how far behind our new model is. Tables 4.5 and 4.6 show the metrics of the two Stanford models. Bar Chart 4.1 compares the precision and accuracy of these four models. The closeness of our projection model with Stanford's dewac model is pretty remarkable. In fact, in three classes, (all except Person) our model outperforms dewac. This result proves the promise of using annotation projection as a compensatory alternative to corpus extension and on par with distributional similarity method of generalization. It also shows that the agreement of annotation in English and German may not be a hundred percent direct; but there is some correlation that can be positively exploited once put into the right setting.

Out of Domain Evaluation

On a closer comparison of precision and recall another pattern of behaviour emerges: the models with projected annotation feature an increase in both TP and FP. This is why precision improves less significantly than recall. Whether or not this could mean that in some cases precision might actually drop is not clear based on this experiment. That is why to further investigate this hypothesis, I run the two CoNLL models on out of domain data as well, that is, the europarl training data. Tables 4.7 and 4.8 show the result of these runs. Here we see a clear drop of precision. Yet, a significant increase of recall too.

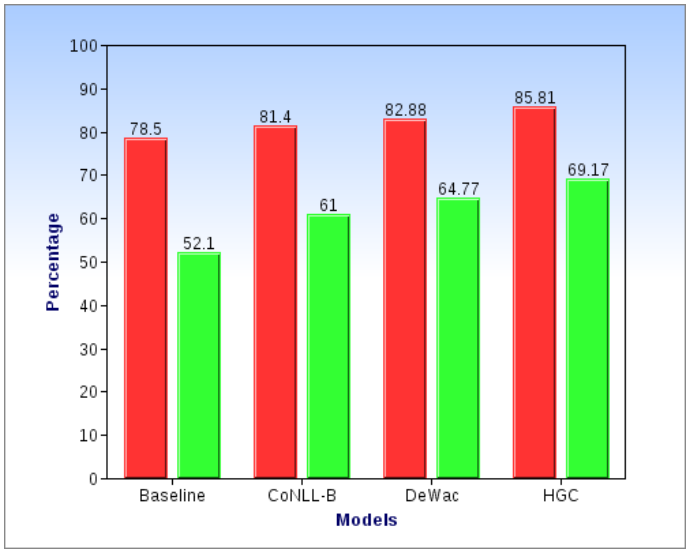

Figure 4.1: Models Performance Comparison (red bar: precision, green bar: recall)

Table 4.7: CoNLL baseline on EuroParl Train

Entity	P	R	F1	TP	FP	FN
I-LOC	93.58	75.0	83.27	102	7	34
I-MISC	58.59	69.05	63.39	58	41	26
I-ORG	88.10	82.22	85.06	74	10	16
I-PER	77.45	68.10	72.48	79	23	37

Therefore, the effect of the projected feature is increase of TP and FP at the same time. For a stronger model (or on more familiar data sets) this still means an absolute increase in both precision and recall, but for weaker models, it might cause a drop in precision, while continues to raise the recall. For many applications of NER, especially those that are concerned with security, recall is a much more important factor, as long as precision remains within an acceptable range.

Table 4.8: CoNLL-B on EuroParl Train

Entity	P	R	F1	TP	FP	FN
I-LOC	77.92	88.24	82.76	120	34	16
I-MISC	51.22	75.0	60.87	63	60	21
I-ORG	76.47	86.67	81.25	78	24	12
I-PER	76.47	86.67	81.25	78	24	12

EuroParl Models Evaluation

Evaluation of these models due to inaccessibility of test data at the time of carrying out the project was postponed to future work.

4.2 Discussion

We tried annotation projection within two different modelling settings: one used the annotation as the target and another as a feature.
Both methods were carried out first on automatically translated corpus with hand annotated named entities, next on parallel corpora with automatically annotated named entities. But the evaluation of the second models were postponed for future work.
In the first method, the model shows worse results than its baseline model. there are two bottlenecks of quality for the A-type model: one is the projection of annotations, the other is translation in one model and source annotations in the other model.
As we saw in the analysis of the generated dataset from the CoNLL translated data, the annotation projection itself cannot be noise free. The translation adds to the noise and ruins the model quality.
All of the works discussed in the Related Work Section are based on this first approach to annotation projection. But a number of incompatible factors, as discussed in that section, makes a direct comparison of results hard.
Yarowsky et al's work [26] is one of the closest ones in methodology to ours; but their training method and their feature set is essentially different from ours. They use a co-training based algorithm and only two features that are also projected form an English parallel corpus. Their target language is also different. The metric they report is the classification accuracy rather than precision, recall, or F-measure. What they report is is very low too compared even to our baseline accuracy; but it is understandable considering the time of the publication. Furthermore, in comparing the metrics, they compare the accuracy of training on the projected annotations with the accuracy of training on the projected annotations and projected features, where quite expectedly the latter gains better results than the former. But they never evaluate the quality of the projected annotations per se or with any other setting, like we do here with the baseline models.
Another work with a similar objective is Ehrmann et al [6] which, as mentioned before, projects annotations from English corpus on to six languages with the purpose building new

generic corpora. However, their projection procedure involves a number of deviations from the more normal path. First, instead of translating the entire corpus, they only translate the named entities, for which they use a combination of SMT and dictionary look-up. Instead of word alignment then, since they only need to match the named entities, they use a combination of String matching techniques and through this process they have to perform manual correction a few times. The major drawback of their work is when they do not offer a clear evaluation of their work due to lack of reference annotations. Rather, they only measure the most accessible metric, which is the recall of projection which only reaches the 90% range when they use all their components in conjunction.

All the hardship involved in this work shows how little trodden the path of annotation projection still is. Even the work of Bentivogli et al [2] which shows great promise for parallel corpora as the source for WSD and other annotated corpus development, involves rigorous manual procedures and evaluations that restricts its usability and make it not easily repeatable.

Based on these comparisons the significance of this work in terms of evaluation is clear. Our model designs allow us an accurate and fast evaluation of training corpus quality without the need for extra resources or manual work.

In the second paradigm, the German corpora, with their German NE-annotation, receive English NE annotations through projection and use them as training features.

This paradigm, applied to the CoNLL data, has the bottlenecks of translation/word alignment from German into English, and the automatic annotation of this translated English text, besides the noise from the projection itself.

The analysis of the merged annotations in Section 4.1.1 showed that the annotation procedure itself can contain a lot of noise mainly due to annotation inconsistency within a hand-annotated corpus and between hand-annotated corpora, especially if they are of different languages.

Chapter 5

Conclusion

In this project we designed two annotation projection paradigms and applied them to two different corpora in German and English: one parallel without Named Entity annotations (EuroParl), the other single corpus with Named Entity annotations (CoNLL 2003).
We used Stanford NER engine and trained four NER models on the training data obtained from the annotation projection paradigms and the feature set of DeWac-generlized and HGC-generalized German NER models developed by Faruqui et al [7] and included as part of the Stanford NER package.
Experiments were carried out to evaluate the two CoNLL models based on their performance improvement over the baseline models (EuroParl evaluation was reserved for future work). Model of the first paradigm, where the projected annotation worked as the training target, showed significant decrease in accuracy. While the second model, which used both the German annotations, as targets, and the English projected annotations, as features, in conjunction, proved significant improvements over the baseline.
The CoNLL Model of the second paradigm even acquired very close results to that of DeWac-generlized NER model that was trained on the same CoNLL corpus but generalized on a large lexicon of German language using Distributional Similarity.
However, in evaluating the paradigm B model on out-of-domain data, the CoNLL model decreased in precision while still improved in recall.
The conclusions and contributions of this project can be summarized as follows:

- This project had a novel approach to annotation projection through its second paradigm: The model trained in this paradigm show an increase of TPs and FPs at the same

time. It means that the model's recall would always increase, while on less familiar datasets the precision might drop. This relatively easy and promising method with a lot of room for improvement can easily be established as a great compensation for data scarcity in different languages.

- This project uses an accurate and fast evaluation method for annotation projection, compared to similar works which either lack proper evaluations or perform it mostly manually. The evaluation metric consists of training a baseline without the projected annotation and comparing the performance of the models on the same testset instead of evaluating the obtained corpus through annotation projection manually.

- Comparing the results of the first paradigm with the second paradigm proves that Using annotation projection in a more controlled way, like as a training feature, instead of placing it the basic target annotation helps the model quality. In principle, a closer looks at projected annotations on German data showed there are many inconsistencies within a hand annotated corpus and between annotated corpora, especially if of different languages, that does not leave the sole act of annotation projection noise-free.

Chapter 6

Future Work

The main recommended directions for future works are as follows:

- We believe the CoNLL-B model is a great model that with some improvement can easily catch up with and push forward the state of the art in German NER. Distributional Similarity features on large lexicon corpora for model generalization are the next best features to incorporate into this model and help it achieve a state-of-the-art quality.
- There is no reason the projected features should be limited to English only: In a many-to-many collaborative fashion with at least all four of the CoNLL languages (since we have the corpora for them) can project their annotations over to the other three languages and receive features from them. This way through a voting based modelling all instances of Named Entities can be guaranteed to be extracted with a high confidence.
- Hand-annotated testsets need to be obtained for EuroParl models in order to evaluate them effectively.
- A lot of the problems in annotating translated data were caused by capitalization and tokenization issues. Using better tools for them can help improving the quality of such models.

Appendix A

Model training configurations

A.1 Training configuration file

A .prop file contains all of the configurations that are required for training. Below is the content of the .prop file we used for training our baseline and paradigm-A models.

```
trainFile = deu.train
serializeTo = deu.train.crf.ser.gz

map = word=0,answer=1

mergeTags = false
useTitle = false
useClassFeature=true
useWord=true
useNGrams=true
noMidNGrams=true
maxNGramLeng=6
usePrev=true
useNext=true
useLongSequences=true
useSequences=true
usePrevSequences=true
```

```
useTypeSeqs=true
useTypeSeqs2=true
useTypeySequences=true
useOccurrencePatterns=true
useLastRealWord=true
useNextRealWord=true
normalize=true
wordShape=chris2useLC
useDisjunctive=true
disjunctionWidth=5
type=crf
useQN = true

# For making faster

QNsize = 10
saveFeatureIndexToDisk = true
maxLeft=1
useObservedSequencesOnly=true
featureDiffThresh=0.05

readerAndWriter=edu.stanford.nlp.sequences.ColumnDocumentReaderAndWriter
```

The "map" feature in the file instructs how the data file should be read, assigning each of the columns a role. In this case, we only have the word at the 0 column and the answer on the 1 column. The "trainfile" and "serializeTo" naturally point to the path of input and output files, respectively. The readerAndWriter line indicates the type of reader that should be used with the input based on data format, which in this case is the general column reader (CoNLL format). Many of the features in this file are self-explanatory. But for more information on them, the documentation of the NERFeatureFactory class of the Stanford engine can be checked.

For paradigm B model, the only difference was an additional label to the map and an

additional feature to the list, as below:

```
map = word=0,tag=1,answer=2

useTags=true
```

□

Bibliography

[1] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.

[2] L. Bentivogli and E. Pianta. Exploiting parallel texts in the creation of multilingual semantically annotated resources: the multisemcor corpus. *Nat. Lang. Eng.*, 11(3):247–261, September 2005.

[3] Andrew Eliot Borthwick. *A maximum entropy approach to named entity recognition.* PhD thesis, New York, NY, USA, 1999. AAI9945252.

[4] Kenneth W. Church and Eduard H. Hovy. Good applications for crummy machine translation. machine translation, 1993.

[5] Veronica Dahl, Sara Saghaei, and Oliver Schulte. Parsing medical text into de-identified databases. 1st International Workshop on AI Methods for Interdisciplinary Research in Language and Biology (BILC), part of ICAART 2011., 2011. In conjunction with the 3rd International Conference on Agents and Artificial Intelligence - ICAART 2011.

[6] Maud Ehrmann, Marco Turchi, and Ralf Steinberger. Building a multilingual named entity-annotated corpus using annotation projection. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, and Nicolas Nicolov, editors, *RANLP*, pages 118–124. RANLP 2011 Organising Committee, 2011.

[7] Manaal Faruqui and Sebastian Padó. Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrucken, Germany, 2010.

[8] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[9] Ralph Grishman and Beth Sundheim. Design of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, MUC6 '95, pages 1–11, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

[10] R. Hwa, Philip Resnik, and Amy Weinberg. Breaking the resource bottleneck for multilingual parsing. 2005.

[11] Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. Evaluating translational correspondence using annotation projection. In *ACL*, pages 392–399, 2002.

[12] Nancy Ide and Tomaz Erjavec. Sense discrimination with parallel corpora, 2002.

[13] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 1 edition, 2000. neue Auflage kommt im Frhjahr 2008.

[14] Kevin Knight. Automating knowledge acquisition for machine translation. *AI Mag*, pages 81–96, 1997.

[15] Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT.

[16] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[17] 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.

[18] Lillian Lee. Measures of distributional similarity. In *Proceedings of the ACL*, pages 25–32, 1999.

[19] Adam Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3), September 2008.

[20] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[21] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[22] F. J. Och and H. Ney. Improved statistical alignment models. pages 440–447, Hongkong, China, October 2000.

[23] Sebastian Padó and Mirella Lapata. Cross-lingual annotation projection of semantic roles. *J. Artif. Int. Res.*, 36(1):307–340, September 2009.

[24] Erik F. Tjong Kim Sang, Sabine Buchholz, and Kim Sang. Introduction to the conll-2000 shared task: Chunking, 2000.

[25] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

[26] David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora, 2000.