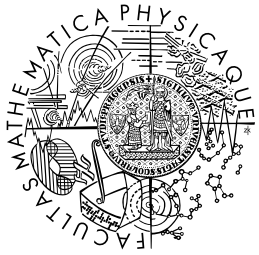

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Vojtěch Srdečný

Talk-Level Domain Adaptation of Speech Recognition

Institute of Formal and Applied Linguistics

Supervisor of the bachelor thesis: doc. RNDr. Ondřej Bojar, Ph.D.
Study programme: Computer Science
Study branch: Programming and Software Systems

Prague 2020

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I thank my supervisor, doc. RNDr. Ondřej Bojar, Ph.D. for his invaluable insight, time and advice, as well as my colleagues at ÚFAL for their helpfulness and patience.

I also thank my family for their support during my studies.

Title: Talk-Level Domain Adaptation of Speech Recognition

Author: Vojtěch Srdečný

Institute: Institute of Formal and Applied Linguistics

Supervisor: doc. RNDr. Ondřej Bojar, Ph.D., Institute of Formal and Applied Linguistics

Abstract: This thesis explores the idea of talk-level domain adaptation for automatic speech recognition (ASR) and machine translation (MT) systems. A quick overview of an existing ASR domain adaptation method is provided. A method for MT domain adaptation is proposed, using an unsupervised MT system. A metric to evaluate the quality of the adaptation process is proposed. The domain adaptation was used on the unsupervised MT system for five different domains. The results of the domain adaptation process are presented and discussed.

Keywords: automatic speech recognition domain adaptation machine translation

Contents

1. Introduction

This work explores two domain adaptation tasks: automatic speech recognition (ASR) and machine translation (MT). By domain adaptation we mean the possibilities of enhancing the performance in a narrow context of usage.

In this chapter, we briefly introduce both of the tasks, then describe our goal of domain adaptation in a closer detail and present the structure of the thesis.

1.1 Automatic speech recognition

ASR (automatic speech recognition) is a process that aims to automatically transcribe spoken language into text. The techniques employed to achieve this goal originate mostly from the computational linguistics field. Examples of ASR usage include generating subtitles for a video or in SLT (spoken language translation), where it is used in conjunction with MT (machine translation). Virtually all ASR systems use *supervised* learning methods, meaning they require an annotated corpus containing audio files and their transcriptions.

ASR systems convert speech in the form of speech signal (a sound wave) into words. To do so, this speech signal is split into a sequence of phonemes, which are the smallest individual units of sound. To convert the sequence of phonemes into individual words, a number of components is employed, one of them being the acoustic model. This model, roughly speaking, contains a mapping between sequences of phonemes and words. Based on this list, it tries to determine the most likely word for a given sequence of phonemes. This list of words is also the complete set of words the system can output. The acoustic model is usually trained using a large *annotated corpus*, which contains of audio files and their transcriptions.

Some ASR systems also contain a language model. Based on the sequence of already recognized words, this model tries to predict words that are likely to come next. This is especially useful when there is a sequence of phonemes, corresponding to multiple similar-sounding words, each with different meaning. This information from language model can help the ASR to pick the correct word. However, when a word is misrecognized, it can have a negative effect on language model's performance, because it can no longer accurately predict what word will come next. The language model is usually trained using a large mono-lingual corpus.

1.2 Machine translation

Machine translation is a task of automatically converting a text from one language to another. Generally speaking, machine translation (MT) models use either *supervised*, *semi-supervised* or *unsupervised* training methods.

Supervised training requires a *parallel* corpus, which consists of pairs of sentences. Each sentence pair has a *source language* half and *target language* half. Essentially, both corpora are a translation of the other, where one side was typically created by translating the other one. By having the exact same sentence

represented twice in both languages, a new information channel opens up. This allows the system to infer additional information from the structures of corresponding sentences, as well as the contexts in which words are used.

Unsupervised training drops the requirement of the corpus being parallel and simply requires having two *monolingual* corpora, which consist of texts in only one language. Unsupervised techniques usually use various methods to make up for this loss of information channel. This type of training is a very recent one.

Semi-supervised training is a combination of supervised and unsupervised training, as it utilizes both parallel and monolingual corpora, essentially having the best of both words.

There are multiple training methods for MT models, but the modern ones include *neural machine translation (NMT)* and classical *statistical machine translation (SMT)*[1].

NMT models are currently one of the most popular ones. They utilize various forms of neural networks. One of the most popular NMT models is the *Transformer* model, which exploits the self-attention mechanism to parallelize the learning process, achieving state-of-the-art results [2, 1].

SMT models are usually phrase-based. One such example is Moses, which utilizes a large amount of parallel data to learn a statistical model, which is then used for phrase-by-phrase translation.

Note that the training methods and model types are orthogonal to each other and can be mixed freely. Unsupervised NMT models generally perform better than others, at the cost of increased computational requirements.

1.3 Talk-level domain adaptation

The slightly ambiguous phrase "talk-level domain adaptation" refers to situations where the speaker and talk topics are known beforehand and the ASR and MT models are tasked with transcribing and translating the talk, respectively. Examples include a lecture at the university or parliament sessions. A domain in this context can be formalized simply as the set of words and phrases common in some field –for example, a lecture about linear algebra is going to contain *domain-specific* words, such as *vector* or *determinant*.

However, ASR and MT models are usually trained on a large amount of *general* data, which might not contain the *domain-specific* words at all. This usually leads into poor performance, as the systems are tasked with transcribing or translating a word they have never seen before.

Thus, the principal idea is to gather domain-specific data for a given *talk*, such as a textbook for a lecture, slides for a presentation or even an article about the given topic. These texts, hopefully containing the *domain specific* words, are then converted into a corpus the ASR and MT models use for their training.

1.4 Discussion topics

In Chapter 2, an existing approach for ASR domain adaptation is discussed, along with the summary of the results. Then, a pipeline for MT domain adaptation is proposed in Chapter 3. Evaluation of the pipeline and data used for this

evaluation are discussed in Chapter 5 and Chapter 4, respectively. Finally, results are shown in Chapter 6 and discussed in Chapter 7.

2. ASR domain adaptation

As the title of this thesis suggests, the original main goal of the thesis was the adaptation of ASR. After the official assignment, this topic was however reasonably satisfactorily handled in master thesis of Jonáš Kratochvíl, as we shortly summarize in this chapter. To avoid duplicating work, we moved towards the topic of MT model adaptation which was a planned extension of the work.

Following is a very short summary of the domain adaptation process explored in the work of Kratochvíl [3] using Kaldi,[1] an ASR framework. The adaptation was performed in two steps: acoustic adaptation and language model adaptation.

2.1 Acoustic adaptation

For acoustic adaptation, the acoustic model was trained using additional in-domain recordings and their transcripts. The recordings were usually of the same speakers as those in the test set used in evaluation later on [3].

2.2 Language model adaptation

For practical reasons, the language model cannot contain each and every possible word. Indeed, many languages, so called *synthetic languages*, have the ability to derive multiple words from a single root by inflection or agglutination. The former is a process of adding morphemes to a root word and the latter is a process of combining multiple morphemes into a single word. Storing all possible word forms can be infeasible, in many cases. Rather, only the most-frequent subset of words is stored, often just the root forms. Additionally, the language model needs to know the context in which a word can occur.

Domain adaptation of the language model was then performed by collecting domain-specific texts into a corpus used for training the language model. First, available domain texts were collected and using sentence embeddings, other corpora were searched for similar sentences. Using these sentences, domain-specific words were identified and again used to search other corpora for sentences containing those words. These sentences formed a corpus used for training the language model. [3]

2.3 Results

The results of the domain adaptation can be seen in Figure 2.1 and Figure 2.2. The relevant parts is the "Level 2" model, in which the language model adaptation was used and "Level 3" model, which extends the "Level 2" model by additionally using acoustic adaptation. The baseline used was trained on a general corpus without any adaptation.

Figure 2.1 shows the *DWA* scores of the models. *DWA* score is a ratio of the recognized domain words to the total amount of domain words. The "Level 2"

[1] https://kaldi-asr.org/

model showed an improvement over the baseline. The "Level 3" model, however, showed an regression over the baseline, as the model "overfitted to the target data extensively" [3].

Figure 2.2 shows the count of out-of-vocabulary words for each domain. The language model adaptation managed to recognize almost half of the overall out-of-domain word count, compared to the baseline. This indicates the chosen method of collecting an in-domain corpus for the language model adaptation can yield good results.

Model type	**EP**	**CL**	**ČRO**	**PS**	**SAO**	**Average DWA**
Domain words	801	802	428	849	236	3116
Google Cloud	81.27	75.68	83.64	81.86	92.37	81.16
UWB	71.78	73.56	92.75	80.44	81.77	78.24
Baseline online	86.14	86.15	96.96	87.63	88.40	88.54
Baseline offline	88.26	86.78	97.66	88.45	88.55	89.24
Level 1	88.01	86.78	96.26	88.69	89.83	89.15
Level 2	88.26	88.52	98.13	89.39	96.61	90.62
Level 3	80.23	85.97	89.33	87.91	94.14	86.21
Level 4	90.76	88.65	98.89	89.63	96.61	91.33

Figure 2.1: DWA results for the five test sets. The numbers indicate the percentage of recognized words from the total of domain words for each test set denoted in the first row. Reproduced from [3].

Model type	**EP**	**CL**	**ČRO**	**PS**	**SAO**	**Total**
Domain words	801	802	428	849	236	3116
Google Cloud	-	-	-	-	-	-
UWB	-	-	-	-	-	-
Baseline online	19	70	0	0	8	97
Baseline offline	19	70	0	0	8	97
Level 1	15	54	0	0	8	77
Level 2	14	24	0	0	6	44
Level 3	14	24	0	0	6	44
Level 4	14	24	0	0	6	44

Figure 2.2: Counts of out-of-vocabulary words in model lexicon. Reproduced from [3].

3. Machine translation domain adaptation

The original assignment of this thesis planned two targets: the primary one was ASR adaptation which was however reasonably well tackled by [3] as described in Chapter 2, and the foreseen extension was MT adaptation. The latter thus became the main focus of the thesis and we outline it in this chapter.

3.1 Introduction and goals

To train a MT system that is aware of a certain domain, the source and target corpora it is trained on have to be related to that domain. The availability of domain-related texts can be quite limited, if the domain is a very specific one, or the language does not have many users. Furthermore, a language's widespreadness does not imply the existence of domain-related texts in the language. For example, English is the de-facto standard language when it comes to academia and research, so obtaining domain-related texts in relatively widespread languages such as German can still prove to be untrivial. Obtaining parallel texts is even more difficult, so this is why we do not assume their existence and use an unsupervised MT system in the pipeline instead.

However, some attempts to sidestep this issue by generating a pseudo-parallel in-domain corpus have been already performed, such as the work of Hu et al. [4], explained in Section 3.4.

Our unsupervised MT system used, Monoses, is trained on corpora generated from texts scraped from the Internet. One of the main weaknesses of unsupervised MT systems can be "a large number of randomly mistranslated named entities which leave a significant impact on the perceived translation quality" [5]. One of the goals of this work is then to check whether or not the proposed approach has the same issue.

3.2 Corpus collection

As mentioned above, in-domain source and target texts can be very scarce. However, for the training of the Monoses model to successfully finish, each of the source and target in-domain corpora have to be sufficiently large. From empirical observations, the minimum required size of a training corpus for this purpose is about 25 to 50 thousand sentences for both source and target sides. In Section 3.2.1 we propose a method for obtaining an in-domain corpus and in Section 3.2.2 we discuss a method to extend this in-domain corpus with sentences from a parallel out-of-domain corpus, if the in-domain corpus is too small.

3.2.1 Mono in-domain corpus

The process of collecting an in-domain corpus is split into multiple phases, as can be seen in Figure 3.1.

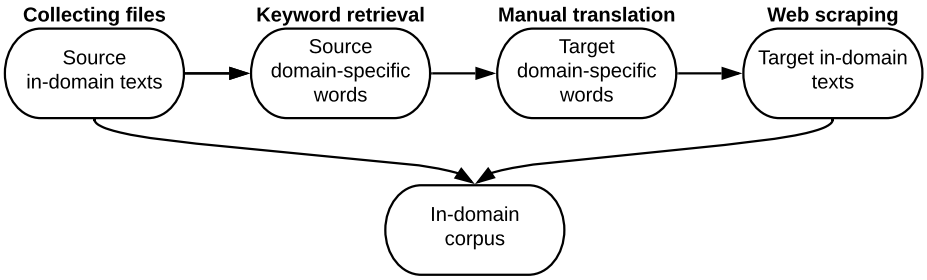

Figure 3.1: Process of collecting the in-domain corpus

Collecting files

As one of the means to collect in-domain files in source language, we modified an already existing application called files-collector.[1] It is a simple Python application based on *Flask*[2] framework that allows file uploading and retrieval. When a need for domain adaptation arises, such as a conference, the app can be easily deployed using Docker's *docker-compose*[3] utility to allow individual speakers to upload their slides or presentation notes.

We then implemented a pipeline for automatic processing of the uploaded files to plaintext using appropriate utilities, such as *pdftotext*[4] for PDF files or *Tikal*[5] for files produced by the Microsoft Office family of programs.

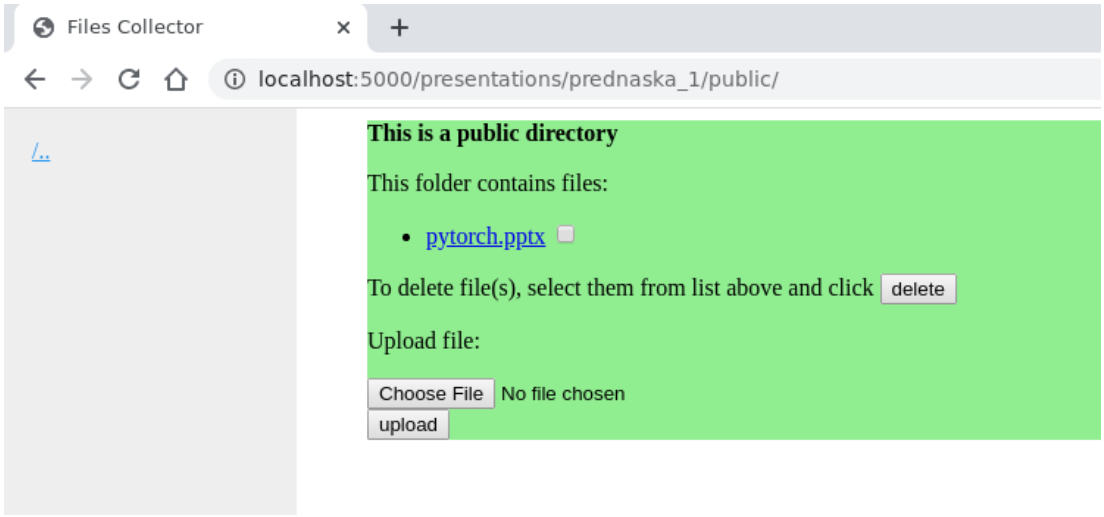

Figure 3.2: Screenshot from the files-collector app

[1] https://github.com/ELITR/files-collector/tree/dockerized
[2] https://palletsprojects.com/p/flask/
[3] https://docs.docker.com/compose/
[4] https://www.xpdfreader.com/index.html
[5] https://okapiframework.org/wiki/index.php/Tikal

Keyword retrieval

To gather in-domain texts in the target language, a list of keywords characterizing the domain is required. Obtaining these keywords can be done in multiple ways, but one of the simpler methods is to compute the most relevant words using *tf-idf*.[6] For each unique token occurring in the text files, *tf-idf* score is computed for each text file. A list of keywords is then obtained by averaging the *tf-idf* scores. This list of is then manually translated to the target language. This could be automated, for example by checking a dictionary made from a large parallel corpus. Note that we do not expect to be able to find translations for all the source keywords. If no translation is found, the keyword is ignored, with the hope that the relevant target-language texts will be found using other keywords and that they will offer translations also for the now unknown words.

Web scraping

The list of domain keywords in target language is then used to scrape the Internet for in-domain texts in target language. We propose to utilize Semantic Scholar[7], a search engine for academic papers. There are alternative search engines, such as Google Scholar.[8] Unlike Semantic Scholar, it does not allow for easy search result scraping. However, Semantic Scholar might be a good source of target in-domain texts only for certain domains, because not all domains have academic papers written about them.

Obtained academic papers and texts were then checked whether or not they are in the target language and converted to plaintexts in the same manner as described in Section 3.2.1.

3.2.2 Mixed in-domain corpus

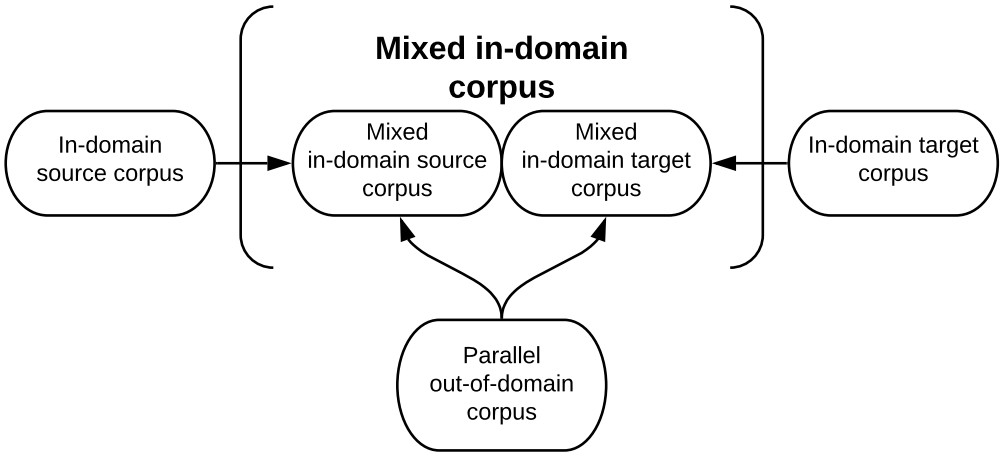

Figure 3.3: Process of generating the mixed in-domain corpus

[6]https://en.wikipedia.org/wiki/Tf-idf
[7]https://www.semanticscholar.org/
[8]https://scholar.google.com/

In some cases, collecting an in-domain corpus of the required minimum size of 25 to 50 thousand sentences can be impractical, if not outright infeasible. For these cases, we propose method of extending the in-domain corpus by out-of-domain sentences from a parallel corpus so it has the required size, as seen in Figure 3.3. More specifically, we take a fixed number of source and target sentences from the parallel out-of-domain corpus and add them to the source and target in-domain corpora. The resulting corpus is called a *mixed in-domain corpus*.

Since Monoses assumes that the input source and target side of the corpus are not parallel, it cannot exploit the fact that the out-of-domain part actually is parallel. To benefit from this parallelism at least to some extent, the cross-lingual word embeddings mapper, *VecMap*[6], is used during training in a *semi-supervised* mode and supplied with a *seed dictionary*, which is a dictionary of source to target words. This helps *VecMap* to generate more accurate cross-lingual mappings.

Seed dictionary

The seed dictionary can be obtained by, for example, utilizing Moses to generate a *lexical translation table*[9] (Figure 3.4). Internally, *MGIZA*,[10] a multi-threaded implementation of *GIZA++* [7], a tool for *word alignment* is used. Given two source and target sentences, *word alignment* is a mapping between words from those sequences. Generally speaking, this mapping is sequence-to-sequence, because a source word can correspond to multiple target words and vice versa. This table can be then converted into a dictionary by simply picking the most probable translation for each source word.

```
europe europa 0.8874152
european europa 0.0542998
union europa 0.0047325
it europa 0.0039230
```

Figure 3.4: Example of a *lexical translation table* used for creating the *seed dictionary*.[11]

3.3 MT model training

Figure 3.5 shows a top-level overview of the training process. The resulting MT model is eventually not used as such. We only extract word translations from it, as discussed in the following chapters.

3.3.1 Monoses

The unsupervised statistical MT system (SMT) used in the training pipeline is Monoses[12] [8]. It uses Moses [9], which is a supervised statistical MT system

[9] http://www.statmt.org/moses/?n=FactoredTraining.GetLexicalTranslationTable
[10] http://www.statmt.org/moses/?n=Moses.ExternalToolsntoc3
[11] http://www.statmt.org/moses/?n=FactoredTraining.GetLexicalTranslationTable
[12] https://github.com/artetxem/monoses

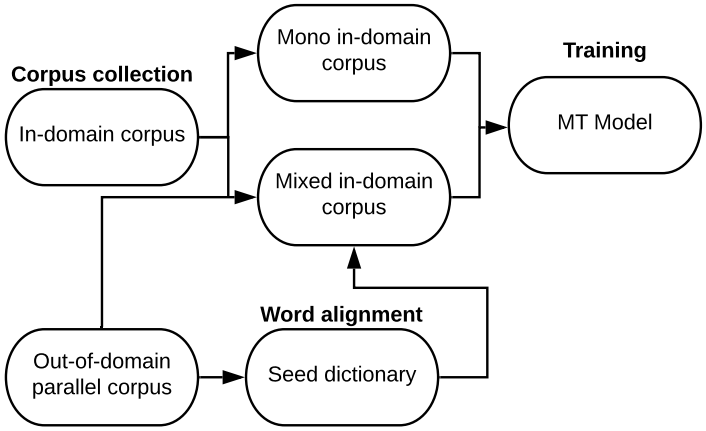

Figure 3.5: Process of MT model training

operating on parallel data. To avoid this need for parallel data, Monoses learns *n-gram word embeddings*, mappings of n-grams to vectors of real numbers, a process popularized by Mikolov et al. [10], using *Phrase2Vec*.[13] Then, these n-gram embeddings are mapped to a shared cross-lingual space using *VecMap*.[14] With this cross-lingual space mapping, a *phrase table* is then induced and used to train a Moses MT model, with the phrase table essentially serving as a phrase-to-phrase source-to-target dictionary.

In a process called *iterative backtranslation*, "a synthetic parallel corpus is created by translating the monolingual corpus in one of the languages with the initial system, and train and tune a standard SMT system over it in the opposite direction" [8]. This process is then usually repeated until the internal weights of the MT model *converge*.

3.3.2 Phrase tables

As mentioned, one of the main components used in the pipeline is a phrase table (Figure 3.6). Each line of the table contains a phrase in the source and target languages, as well as probabilities of the translation and alignment information.

```
ZABRÁNIT REGISTRACE K DPH ZAVÁDÍ | WIRD MIT EINEM KLEINEREN | 1 | 0-0 1-1 2-2 3-2 4-3
('prevent registration for VAT introduces') | ('will with a smaller')
ZABRÁNIT REGISTRACE K DPH | WIRD MIT EINEM | 0.333 | 0-0 1-1 2-2 3-2
('prevent registration for VAT') | ('will with a')
ZABRÁNIT REGISTRACE | WIRD MIT | 0.5 | 0-0 1-1
('prevent registration') | ('will with')
```

Figure 3.6: Excerpt from the initial phrase table induced by Monoses from non-parallel data, so the accuracy is very low. English gloss is in quotes.

The final phrase table generated by the iterative backtranslation process will later on be used to evaluate the accuracy of the MT model, as described in Chapter 5.

[13] https://github.com/artetxem/phrase2vec
[14] https://github.com/artetxem/vecmap

3.4 Related work

The *mixed in-domain corpus* method Section 3.2.2 was inspired by the DALI [4] (Domain adaptation by lexicon induction) method for unsupervised MT. (Figure 3.7). It also uses *GIZA++* to extract a supervised *seed dictionary* from a large parallel general corpus. Then, an unsupervised *seed dictionary* is generated using a generative adversarial network (GAN), a method based on neural networks. Then, a mapping between these two seed dictionaries is learned and used to create a pseudo-parallel in-domain corpus which is then used for supervised training of neural-based MT systems.

The main difference between our proposed method and the DALI method is that we do not generate the mapping W^*. The supervised seed lexicon is generated by MGIZA in the form of the seed dictionary (Section 3.2.2). We hope Phrase2Vec and VecMap used by Monoses will produce a sufficient mapping instead by inducing the initial phrase table and using the backtranslation process, to eventually create a pseudo in-domain parallel corpus.

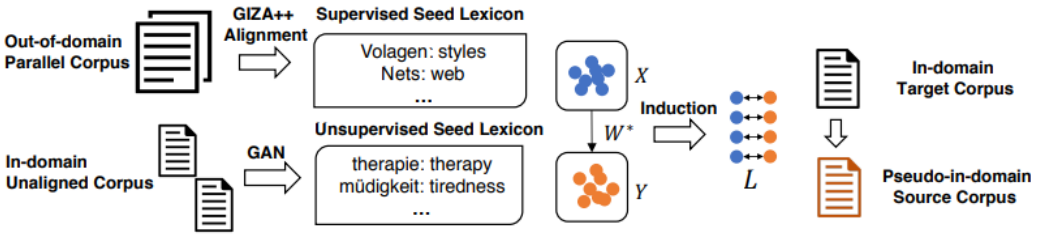

Figure 3.7: Work flow of the DALI method. [4]

4. Training data

Languages chosen to evaluate the performance of the pipeline were Czech and German, with Czech being the *source* and German being the *target*.

4.1 Czech domain-specific corpora

In order to extend Kratochvíl's work [3] and possibly compare the domain adaptation impact on ASR and MT, the five domains evaluated in [3] are also used for evaluation of our proposed pipeline:

The Český Rozhlas (ČRO) corpus consists of transcriptions of shows and programmes broadcasted on the Czech radio station Český Rozhlas. The Czech Parliament (PS) corpus consists of transcriptions of plenary hearings. Computational linguistics corpus (CL) contains transcriptions of two talks by Martin Popel, both focused on machine translation. European Parliament corpus (EP) consists of transcriptions of plenary hearings that took place in the European Parliament. Finally, the Supreme Audit Office corpus (SAO) consists of transcriptions of talks given at a SAO conference.

Each corpus presents a challenge. Their size varies wildly, as well as the range of topics discussed – some corpora focus on only one topic (CL, SAO) while others contain multiple subtopics (PS, ČRO, EP). Their size is in Table 4.1.

Corpus	**# of words**
PS	5510121
ČRO	2710811
CL	65783
EP	131151358
SAO	1268476

Table 4.1: Overview of Czech domain-specific corpora sizes

4.2 German domain-specific corpora

Because of the small count of domains, the keywords (introduced in Section 3.2.1) used for scraping were picked and translated manually. Texts were scraped from the Semantic Scholar site. The size number of keywords used for scraping and the size of resulting corpora is in Table 4.2.

4.3 Parallel corpus

The Czech-German parallel general corpus (PG) used for training the baseline model and for generating the *seed lexicon* is a combination of multiple corpora from OPUS[1] [11]. The corpora used are DGT, EMEA, EUbookshop, Europarl,

[1] http://opus.nlpl.eu/

Corpus	# of keywords	# of words
PS	5	425430
ČRO	3	684813
CL	2	319093
EP	5	367438
SAO	2	302664

Table 4.2: Overview of German domain-specific corpora sizes

JRC-Acquis, MultiParaCrawl, News-Commentary, ECB and OpenSubtitles. The size of the parallel corpus is in Table 4.3.

Corpus	# of sentences
PG	21373550

Table 4.3: Overview of parallel corpus size

5. Evaluation

The goal of this thesis is specifically the adaptation of MT to new domains. Unlike in unsupervised MT where the goal is to learn to translate everything, complete sentences, from monolingual texts, we assume that the common words and expressions will be covered by the baseline system, and that only the *domain-specific* terms will cause an issue. For this reason, we do not use any standard MT evaluation measures such as BLEU[1] score, because they assess the complete sentences and they would be influenced by the few domain-specific words too little. Instead, our evaluation score checks the extent to which domain-specific terms are automatically found by our approach.

The proposed evaluation metric is based on matching a manually created reference dictionary of domain-specific words with a dictionary that is automatically constructed from the final phrase table generated by the unsupervised MT model.

5.1 Domain-specific words

For the purpose of evaluation, the lists of source domain-specific words used by Kratochvíl [3] were reused. For example, the domain-specific words for the SAO domain include *reverse charge,*[2] *plátce daně, registr, DPH, EET, OSS, audit, transakce*. Notably, these lists can contain multiple forms of the same *lemma*, a base form of a word, such as *registrovaní, registrovaná* in the SAO domain. While listing multiple word forms is a sensible decision for speech recognition where the goal is to find exactly the form which was uttered, in MT we need to see if the lexical counterpart was identified acceptably, regardless its exact form. To avoid evaluating the same word twice, the domain-specific words were lemmatized using MorphoDiTa [12], "an open-source tool for morphological analysis".[3] An overview of the number of domain-specific words before and after lemmatization is in Table 5.1

Corpus	**# of words**	**# after lemmatization**
PS	106	87
ČRO	48	40
CL	129	88
EP	111	76
SAO	63	49

Table 5.1: Overview of domain-specific word list sizes

5.1.1 Reference domain translation dictionary

With the list of source domain-specific words, we need to obtain their translations. To obtain a reference Czech to German dictionary of the domain-specific

[1] https://en.wikipedia.org/wiki/BLEU

[2] this English word is actually a jargon term used in Czech spoken text

[3] http://lindat.mff.cuni.cz/services/morphodita/

words, services of a translation agency were utilized. Given the linguistic differences between Czech and German, asking to create a word-for-word dictionary is not very sensible. For example, in Czech, declensions are identified by different word endings, to a large part. However, in German, declensions are identified by a combination of articles, nouns and sometimes even the context of the sentence. Furthermore, values of morphological categories often do not correspond to each other across languages. Gender and case can easily differ because they depend on the noun or on the governing verb and its valency frame. The lexical correspondence is also not straightforward, multi-word expressions in one language can correspond to a single word on the other one. This is particularly common for German compounds where some of their components are translated, for example, into adjectives or other parts of speech in more complicated cases. Thus, the translators were asked to supply all possible translations of a given domain-specific word with respect to form and domain context, including articles and nouns, if possible.

```
ZPRAVODAJI 'to/via the reporter/s' : (DER) REPORTER, (DER) BERICHTERSTATTER, (DURCH DIE) REPORTER
ZPRAVODAJ 'reporter' : (DER) REPORTER, (DER) BERICHTERSTATTER
GARANČNÍMU 'guarantee' : (DEM) GARANTIE-
```

Figure 5.1: Example from the Czech-German hand-crafted reference dictionary of domain-specific words. English gloss is in quotes and does not appear in the dictionary.

As can be seen in Figure 5.1, different forms of a single lemma, in this case *'ZPRAVODAJ'*, can have multiple appropriate translations, some of them with one or more articles and prepositions. Some of the words, especially the adjectives, such as *'GARANČNÍMU'* are translated as a prefix, denoted by the dash at the end of the translation: *'GARANTIE-'*.

This hand-crafted dictionary was then further manually processed. First off, translations of all forms of a given lemma were consolidated together. Second, some German translations consists of multiple words, but the focus is on comparing word-for-word translations. To accommodate for this, translations with multiple words were included twice – once with all verbs and nouns, and once with just the base word. More thorough reasoning for this second step is in Section 5.4.

The resulting dictionary is called the *golden dictionary* and can be seen in Figure 5.1. Essentially, it represents a list of all possible translations in the target language of a given lemma in the source language. Notably, this list of translations is by no means complete, as it will contain only hand-picked translations of forms of lemmas present in the list of *domain-specific* words. Gathering a list of all possible translations of all lemma forms would be too time consuming and expensive. This deficit was compensated for by allowing an inexact match when comparing the translations. This is discussed in Section 5.4.1.

5.2 Phrase table dictionary extraction

Each line in the phrase table contains the source and target phrase, probability of translation and word alignments, as can be seen in Figure 3.6. Using the word

```
ZPRAVODAJ 'reporter' : DER REPORTER, REPORTER, DER BERICHTERSTATTER,
BERICHTERSTATTER, DURCH DIE REPORTER
GARANČNÍ 'guarantee' : (DEM) GARANTIE-, GARANTIE-
```

Figure 5.2: Examples from Figure 5.1 after being manually processed into a *golden dictionary* of *golden translations*. Gloss is in quotes.

alignment information, the source and phrase phrase pairs are converted into a series of corresponding words, as shown in Figure 5.3. These translations are called *extracted translations* and they form the *extracted dictionary*.

```
PRO DAŇOVÉ ÚČELY  'for tax purposes' |  DER UNTERNEHMENSSTEUER  'corporate tax'
      | 0.58 |  0-0 1-1 2-1
=>
(PRO, DER, 0.58), (DAŇOVÉ, UNTERNEHMENSSTEUER, 0.58), (ÚČELY, UNTERNEHMENSSTEUER, 0.58)
```

Figure 5.3: Example of a line from a phrase table being converted into *extracted translations*, forming the *extracted dictionary*. Line break added for readability. Gloss is in quotes.

5.3 Precision and recall

The two traditional metrics used for word-for-word comparison are *precision* and *recall*.

$$precision = \frac{\#\text{ of true positives}}{\#\text{ of true positives} + \#\text{ of false positives}}$$
$$recall = \frac{\#\text{ of true positives}}{\#\text{ of true positives} + \#\text{ of false negatives}}$$

Figure 5.4: Definition of precision and recall

In layman's terms, precision is the ratio of the number of correct answers to the number of total answers. Recall is the ratio of the number of correct answers to the number of total correct answers. High precision indicates rarely answering wrong, while high recall indicates few omitted answers. This subtle difference can be understood by observing the edge cases: answering nothing at all yields the precision of 1, while providing all possible answers to a query yields the recall of 1. This also implies precision and ratio are complementary to each other in a way, because increasing one will usually decrease the other.

However, because a single source word has multiple golden translations, thus multiple correct answers, using these two methods in a straightforward manner would not yield meaningful results. Thus, two metrics with semantic meanings similar to precision and recall are proposed in the following Section 5.4.

5.4 Metrics

For each lemmatized domain-specific word ω on the left (source) side of the golden dictionary shown in Figure 5.6, two metrics will be calculated. First,

all forms ω_i of the lemma ω were generated using MorphoDiTa. If MorphoDiTa fails to provide forms of the lemma, typically when the domain-specific word is an abbreviation or a name, the form itself is used instead. Then, a list of *proposed translations* ρ_k of each form ω_i was compiled by searching the *extracted dictionary* shown in Figure 5.3. Each *proposed translation* consists of a target phrase and the probability of the translation. A list of *golden translations* τ_j for the domain-specific word ω is also collected using the *golden dictionary* mentioned in Figure 5.6. Finally, *proposed translation score* S_k is calculated for each *proposed translation* ρ_k. An overview of the relationship between these newly introduced objects is in Figure 5.5.

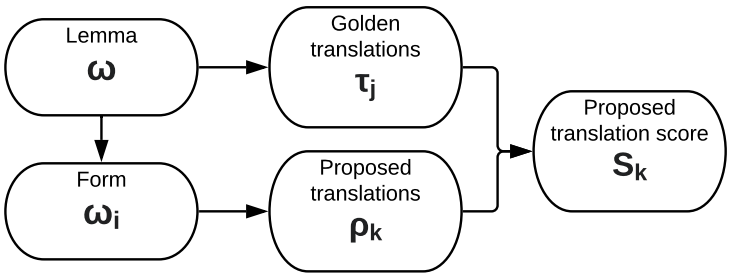

Figure 5.5: Relationship of objects used in the evaluation process

```
INLÄNDISCHEN 'inland': 1,
ERFASSUNG 'registration': 1,
HIER 'here': 1
```

Figure 5.6: Examples of automatically *proposed translations* for the lemma "REGISTRACE" ('registration'). Gloss is in quotes.

5.4.1 Fuzzy matching

The golden translations are only a subset of all possible translations, as they were created by translating only a limited amount of the domain-specific word forms. It is possible for a *proposed translation* to be *correct*[4], but the target phrase of the proposed to not be present in the list of golden translation. If we used strict equality for checking if the translation is correct or not, the proposed translation would be treated as not correct. To avoid this, the matching between *golden translations* and *proposed translations* is performed in a *fuzzy* way, utilizing FuzzyWuzzy, [5] a Python library, which uses Levenshtein distance to calculate the similarity ratio of two strings.

[4]From an objective standpoint.
[5]https://github.com/seatgeek/fuzzywuzzy

5.5 Score definitions

Using the string similarity function $sim(x, y) \mapsto [0, 100]$, representing the similarness percentage, each proposed translation $\rho_k = \textit{(target, probability)}$ of a lemma form ω_i is then assigned a score S_k, iterating k over all golden translations τ_j:

$$S_j = \begin{cases} 100 * probability & \text{if } \tau_j \text{ is a prefix of } target^6 \\ sim(target, \tau_j) * probability & \text{otherwise} \end{cases}$$

Figure 5.7: Translation score formula

Each in-domain lemma ω is then assigned two scores: ω_{max} and ω_{avg}, iterating i over all lemma forms ω_i, j over all golden translations τ_j and k over all proposed translation scores S_k of the lemma form ω_i as described on Figure 5.8.

$$\omega_{max} = max_{i,j,k}\ (S_k)$$
$$\omega_{avg} = avg_i\ (max_{j,k}(S_k) \text{ if } \omega_i \text{ has at least one proposed translation})$$

Figure 5.8: Word score formulas

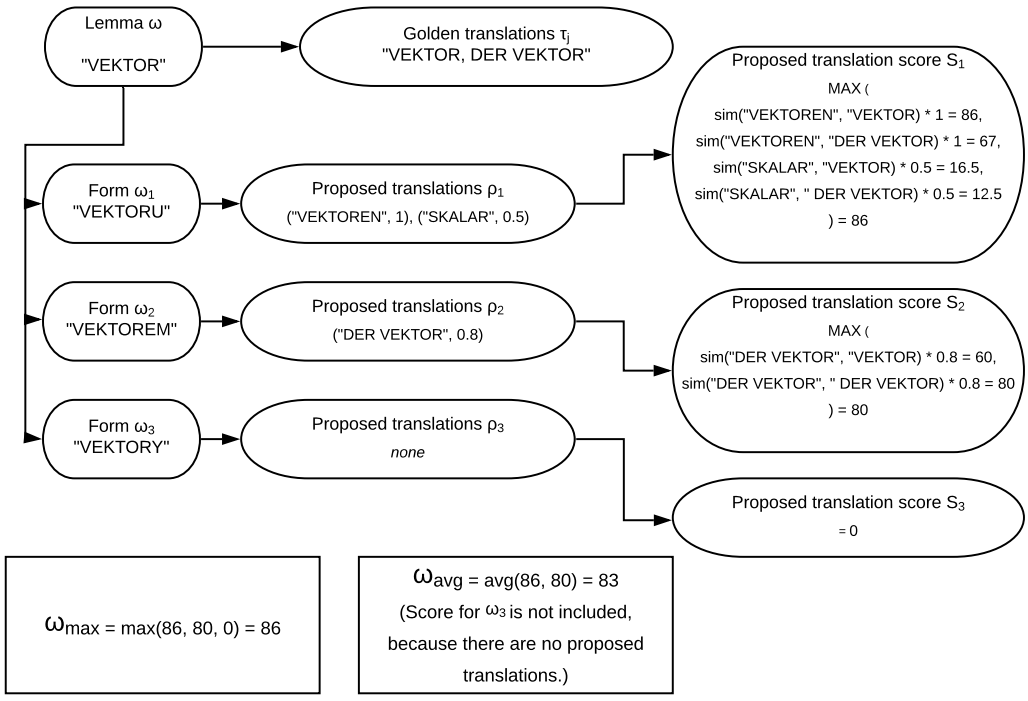

Figure 5.9: Fictional example of computing ω_{max} and ω_{avg}

An example of the computing process is displayed in Figure 5.9. ω_{max} has the semantic meaning of *recall* – whether the phrase table contains the correct translation at all. ω_{avg} represents *precision* – how many forms in the phrase table have an acceptable translation. Because the lemma can have many more forms than those present in the phrase table, lemma variants without a proposed translation are not included in the average, to avoid excessive penalization.

[6] Only if the golden translation is explicitly marked as a prefix.

5.6 Recognized words

Because all forms of a domain-specific word ω are checked when extracting translations, the extracted translation's source phrase is not necessarily in the same form as the original word ω. This can result in a word score ω_{max} that is not *100*, because the target phrase is being compared to a list of golden translations of source forms, that do not have to necessarily contain the extracted translation's source form. In this case, ω_{max} will be lightly smaller, even though the system has recognized the word correctly and provided a translation that is accurate enough.

Naturally, a need for a score cutoff to consider a domain-specific word as *recognized* arises. From manual inspection of the results, translations with the ω_{max} score of *90* and above were very similar to the golden translations, with only one or two differing characters. Obviously, this is only a poor estimate and it could be improved by calculating the average string similarity between all forms of the golden German translation. However, this would require a German lemmatizer and word form generator. Unfortunately, especially the latter wasn't readily available. Because of this reason, only the domain-specific words with a ω_{max} score of *90* or higher are reported as *recognized*.

5.7 Learnable words

Because the source and target corpora aren't parallel, not all words from the source corpus will occur in the target corpus. In this case, the MT system cannot be reasonably expected to infer a translation of a domain-specific word that simply does not appear in the target corpus. The domain-specific words that have at least one form in the source corpus and at least one golden translation on the target corpus are reported as *learnable* words.

5.8 Seed dictionary

The seed dictionary used for semi-supervised training of the cross-lingual word embeddings used to train mixed in-domain models as described in Section 3.2.2 is synthesized from a large parallel corpus, so it can contain translations of the domain-specific words occurring in the corpus. Thus, if a mixed in-domain model recognizes a domain-specific word, it isn't clear if the information needed to infer the translation was contained in the in-domain corpus or in the seed dictionary. To alleviate this, the seed dictionary was compared to the dictionary of golden translations to see which domain-specific words are recognized by the seed dictionary.

Because the seed dictionary is created by word alignment (Section 3.2.2), the entries can contain extraneous characters, typically punctuation (Figure 5.10). To account for this, FuzzyWuzzy (Section 5.4.1) was used for determining if a domain-specific word has a translation in the seed dictionary. For a domain-specific word to be reported as *present in seed dictionary*, the source form has to match the source word in the seed dictionary and at least one golden translation has to be at least 90% similar to the corresponding target word.

```
'PŘEREGISTROVÁNÍ': 'UMREGISTRIERUNG'
'PŘEDREGISTRACI,': 'STORNIEREN,;
'REGISTRACE.': 'REGISTRATION.'
```

Figure 5.10: Example of seed dictionary entries for the source word 'REGISTRACE' ('registration'). Note that the semantic swap in the second line where the target 'STORNIEREN' means 'to cancel' is quite common in phrase-based MT. The negation was somewhere in the sentence but it was not extracted with the (negated or negative) word itself.

6. Results

6.1 Mono in-domain corpora

The following are the results of scores trained solely on in-domain data. Each model took up to 12 hours to train, depending on the corpus size, using 30 threads on a Xeon E5-2630 CPU. Note that training of the CL model failed because of small size of the Czech in-domain corpus, so results for this model are missing. The recognized words and their count are presented in Table 6.1 and Table 6.2.

Model	# recognized	# learnable	# learnable in PG
PS	2	41	63
ČRO	1	28	34
CL	-	68	76
EP	0	58	67
SAO	3	35	43

Table 6.1: # of recognized words for mono in-domain models

6.1.1 Czech Parliament (PS)

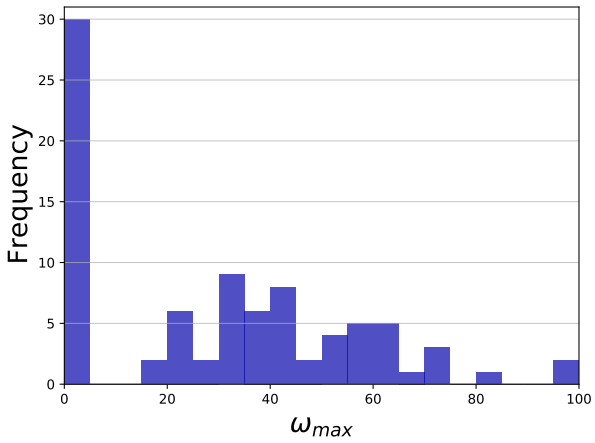

Figure 6.1: Frequency of ω_{max} scores for PS (mono)

The Czech Parliament domain was a very challenging one. The source corpus consists of transcriptions of plenary hearings, so the domain-specific words include names of the many speakers and their political parties. This resulted in many of the words simply not being present in the target corpus, as seen on Figure 6.3. Another consequence is the large number of ω_{max} scores of 0 in Figure 6.1.

The two recognized words are *'ROK' ('year')* and *'MINISTR' ('minister')*. Both have ω_{max} score of 100 and ω_{avg} scores of 61 and 57 respectively, as seen on Figure 6.2. Notably both of those words are one of the most frequent in source and target, as seen on Figure 6.3.

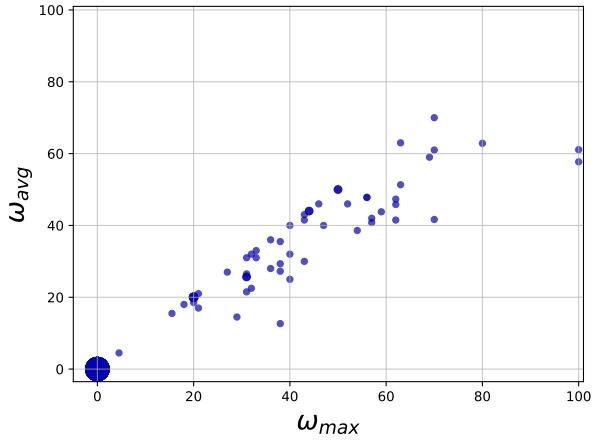

Figure 6.2: Distribution of ω_{max} and ω_{min} scores for PS (mono)

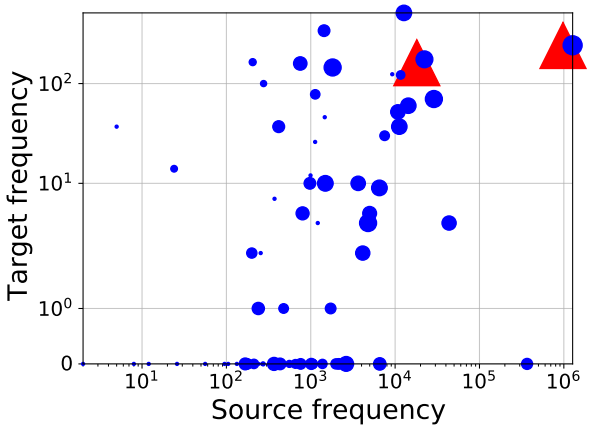

Figure 6.3: Frequencies of PS (mono) domain-specific words in source and target corpora. Size of markers is proportional to ω_{max} score, triangles denote *recognized* words.

6.1.2 Euro Parliament (EP)

The Euro Parliament model performed the worst with zero recognized domain-specific words, as seen in Figure 6.4. ω_{max} score of 0 implies no translations were proposed for the given words, meaning none of the words of interest are present in the phrase table. Figure 6.5 shows the most common count of proposed translations for a given word was one, because the ω_{max} score is the same as ω_{avg}. These results are surprising, because Figure 6.6 shows a lot of words were present both in the source and the target corpus. However, as those words are not present in the final phrase table, indicated by the ω_{max} score of 0, it seems like the iterative backtranslation process mentioned in Section 3.3.1, did not perform well, possibly because the size of the source corpus is much larger than the target corpus.

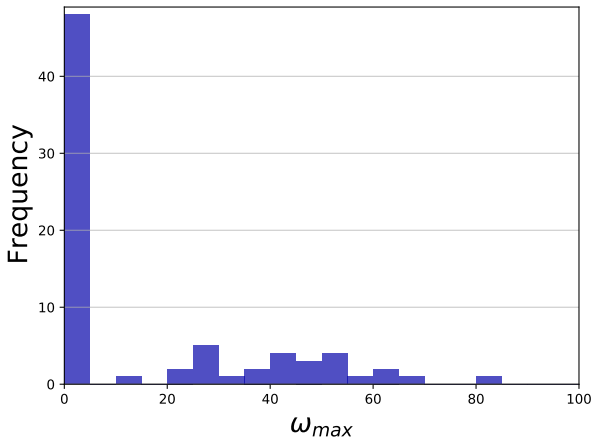

Figure 6.4: Frequency of ω_{max} scores for EP (mono)

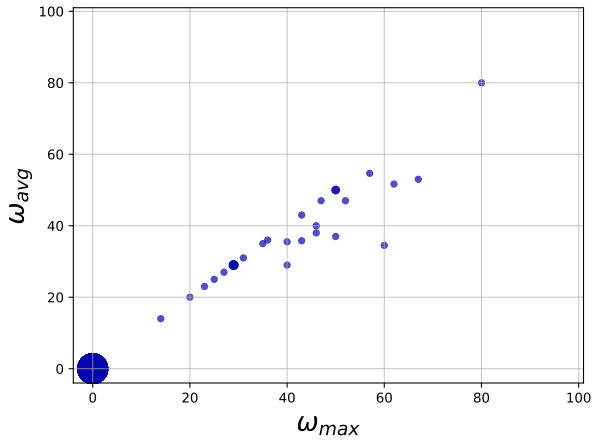

Figure 6.5: Distribution of ω_{max} and ω_{min} scores for EP (mono)

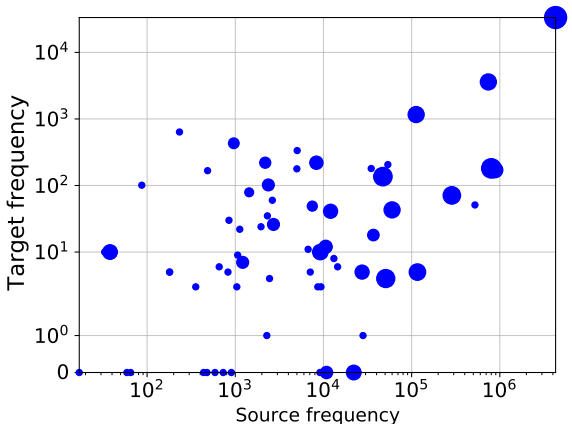

Figure 6.6: Frequencies of EP (mono) domain-specific words in source and target corpora. Size of markers is proportional to ω_{max} score, triangles denote *recognized* words.

6.1.3 Český Rozhlas (ČRO)

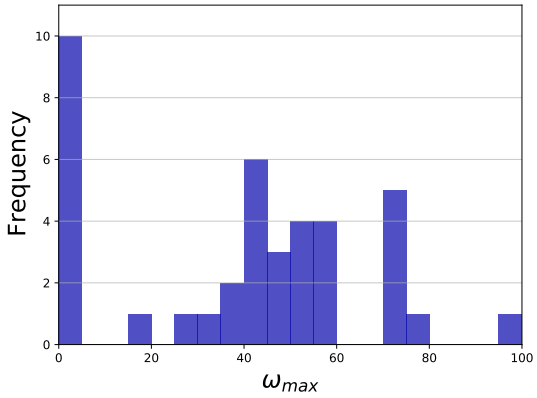

Figure 6.7: Frequency of ω_{max} scores for ČRO (mono)

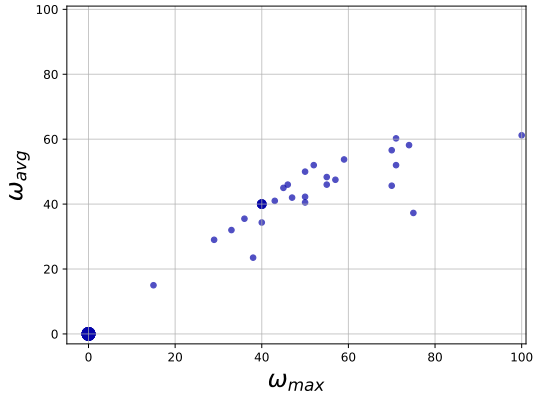

Figure 6.8: Distribution of ω_{max} and ω_{min} scores for ČRO (mono)

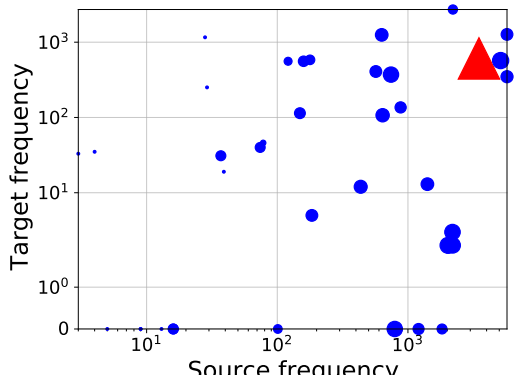

Figure 6.9: Frequencies of ČRO (mono) domain-specific words in source and target corpora. Size of markers is proportional to ω_{max} score, triangles denote *recognized* words.

The ČRO model's performance was average, with one domain-specific word recognized. As Figure 6.7 shows, only comparatively few words had a ω_{max} score of 0, indicating most of the words had at least one proposed translation. The average score is around 50. However, the closest translation being only 50% similar to some golden translation means it is not useful, because the two words are completely different. This performance is a bit disappointing, because Figure 6.9 indicates a lot of words were present both in source and target corpora.

The only recognized word is *'EVROPSKÝ,' ('European')*, with ω_{max} score of 100 and ω_{avg} score of 61 (Figure 6.8). This word is also one of the most frequent in source and target corpora.

6.1.4 Supreme Audit Office (SAO)

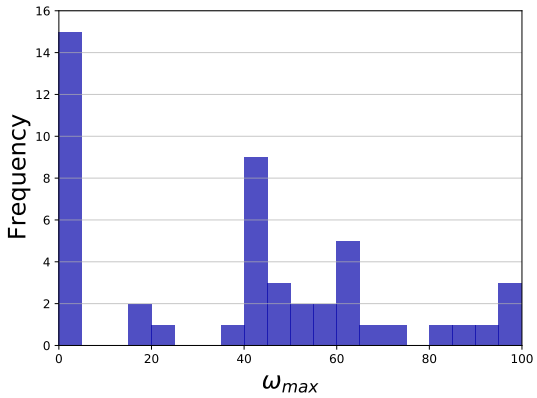

Figure 6.10: Frequency of ω_{max} scores for SAO (mono)

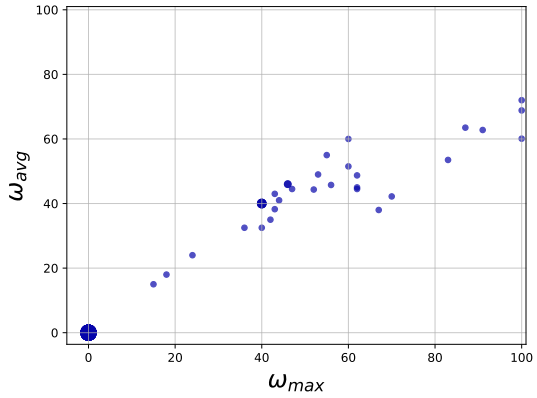

Figure 6.11: Distribution of ω_{max} and ω_{min} scores for SAO (mono)

The SAO model was the best performing one, with Figure 6.10 showing 4 recognized domain-specific words, although with a relatively high number of words with ω_{max} score of 0. This is partially because some of these zero-scoring words are abbreviations or names. Figure 6.12 indicates a good distribution of the domain-specific words in the source and target corpora, which was possibly one of the reasons of the good performance. The recognized words are *'DAŇ' ('tax'), 'DAŇOVÝ' ('taxative'), 'REGISTRACE' ('registration')*. Their ω_{max} scores were

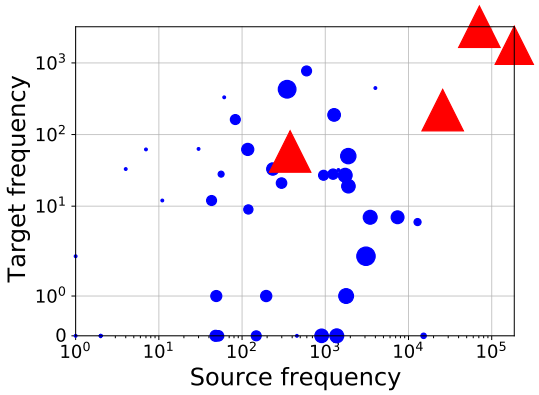

Figure 6.12: Frequencies of SAO (mono) domain-specific words in source and target corpora. Size of markers is proportional to ω_{max} score, triangles denote *recognized* words.

100, 100, 91 and 100 respectively (Figure 6.11). However, their ω_{avg} scores were 68, 60, 62 and 72 respectively, which are one of the highest compared to the recognized words of other models. This indicates multiple forms of these words had proposed translations of a good quality.

Model	**Recognized words**
PS	*MINISTR, ROK*
	'minister', 'year'
ČRO	*EVROPSKÝ*
	'european'
CL	*not evaluated*
EP	*none*
SAO	*DAŇ, DAŇOVÝ, REGISTRACE*
	'tax', 'taxative', 'registration'

Table 6.2: Recognized words in mono in-domain models.

6.2 Mixed in-domain corpora

Because the training of the CL model on mono in-domain data failed, the source and target in-domain corpora were extended by 50 thousand sentences from the parallel corpus (PG).

6.2.1 Seed dictionary

As discussed in Section 3.2.2, the seed dictionary can contain translations of domain-specific words, which can be seen in Table 6.5.

Model	# recognized	Δ recognized	# learnable	Δ learnable
PS	6	+4	43	+2
ČRO	1	0	30	+2
CL	1	+1	81	+13
EP	5	+5	65	+7
SAO	4	+1	39	+4

Table 6.3: # of recognized words for mixed in-domain model. Deltas are with respect to the mono models.

Model	Recognized words
PS	*VLÁDA, MINISTR, BOD, ROK, STRANA, VÝBOR* *'government', 'minister', 'point', 'year', 'party', 'commitee'*
ČRO	*EVROPSKÝ* *'european'*
CL	*BASED* *'based'*
EP	*ČLENSKÝ, DE, TRANSPORT* *'member', 'DE', 'transport'*
SAO	*EVROPSKÝ, REGISTR, DAŇ, FINANČNÍ* *'european', 'register', 'tax', 'financial'*

Table 6.4: Recognized words in mixed in-domain models.

Domain	Present words
PS	*BAUER, BENDA* *'Bauer', 'Benda'*
ČRO	*ANDREJ, TWITTER* *'Andrej', 'Twitter'*
CL	*ATTENTION, AUTOMATICKY, GOOGLE, KONTEXT* *LEARNING, ONLINE, REFERENCE, TRANSFORMER* *'attention', 'automatic', 'Google', 'context'* *'learning', 'online', 'reference', 'transformer'*
EP	*EXPORT, INTER, SANKCE, STREAM, ŽADATEL* *'export', 'inter', 'sanctions', 'stream', 'applicant'*
SAO	*REGISTROVANÝ, REVERS, TEORETICKY* *'registered', 'reverse', 'theoretical'*

Table 6.5: Domain-specific words present in the seed dictionary.

6.3 Discussion

6.3.1 Mono in-domain models

The performance of mono in-domain models is relatively poor, with only three models recognizing any domain-specific words at all. One of the major surprises

is the performance of the EP model, which recognized 0 words, while having the largest source corpus of all domains. The corpus size might have been it's downfall, as discussed in Section 6.1.2.

Another surprise is the performance of the SAO model, which recognized 4 domain-specific words. Three of those words are one of the most frequent in the source and target corpora (Figure 6.12). This, along with the performances of the remaining models, ČRO and PS, suggests that for a word to be recognized, it has to appear in the source and target corpora with a relatively high frequency.

6.3.2 Mixed in-domain models

The sentences from the parallel corpus (PG) added to the in-domain corpora, contain some of the domain-specific words, so an increase in *learnable* words, is expected (Table 6.3), especially when the in-domain corpus was already small (CL). The mixed in-domain models mostly recognize all words as the mono in-domain models, as well as previously unrecognized words. The mixed in-domain model for SAO no longer recognizes the words *'REGISTR', 'DAŇOVÝ'*, although it recognizes the word *'REGISTRACE'*. For example, the ω_{max} score of 'REGISTR' in the mixed model is only 60 and the best proposed translation is 'FRISTEN' ('to lead'). This suggests the impact of including parallel sentences into the in-domain corpora can result in some regressions (words that are no longer recognized). This can be caused by the domain-specific words that appear in the parallel sentences in a different context than the domain one, so the MT model does not translate them correctly, in regard to the domain.

6.3.3 Seed dictionary

The domain-specific words present in the seed dictionary are displayed in Table 6.5. None of the words recognized by the mixed in-domain models are present in this table. This suggests the domain-related words were recognized using the information from the corpora and not by the information in the seed dictionary.

6.4 Type of recognized words

Almost all of the recognized domain-specific words from mono and mixed in-domain models are verbs or nouns. No names such as *'Merkel'* or abbreviations as *'DPH' ('VAT')* were recognized.

7. Conclusion

7.1 Summary

We discussed the basic principles of ASR and MT systems and explored an existing ASR domain adaptation method. We then proposed our MT domain adaptation pipeline, along with metrics to evaluate the success of the adaptation.

The proposed method of obtaining an in-domain source and target corpus yielded a corpus that does contain a high number of source and target domain-specific words. Adding the parallel sentences from a general parallel corpus increased the number of *learnable* words only by a small amount (Section 5.6). The obtained target in-domain corpus was, in some cases (EP), comparatively small with respect to the source in-domain corpus.

However, as seen in Table 6.1, in the mono in-domain corpora there are on average 75% of *learnable* in-domain words, compared to the count of *learnable* words for each domain in the PG corpus. This might look as a poor result, until we compare the target corpora size: the PG corpus has roughly 220 million words, but the in-domain corpora word count ranges from roughly 300 thousand to 700 thousand. This result indicates our proposed method of collecting the in-domain source and target corpora is efficient at obtaining in-domain texts containing domain-specific words.

As discussed in Section 6.3, the models trained on purely in-domain data had a relatively low performance, with only a few domain-specific words recognized at best. Adding parallel out-of-domain data to the in-domain corpora can yield improved results, with more domain-specific words *recognized*. However, it can also result in regressions, where previously recognized domain-specific words are not *recognized* anymore.

Improving the target in-domain corpus method to collect texts from multiple sources could yield better results, as the singular source, Semantic Scholar, is not a good fit for some of the domains. The small corpora size is one of the possibilities the proposed pipeline performed poorly at *recognizing* the domain-specific words, compared to other unsupervised MT systems, such as *DALI*, discussed in the following section Section 3.4.

We hoped by gathering corpora with a large amount of *learnable* words, we could achieve better results when *recognizing* the domain-specific word. This is an area where traditional unsupervised SMT methods struggle, as described in Kvapilíková et al. [5]. Unfortunately, our pipeline also struggled with *recognizing* the domain-specific words.

Overall, our pipeline managed to efficiently collect in-domain corpora with a comparatively large count of *recognized* in-domain words. Unfortunately, it struggled with *recognizing* the in-domain words, but this might be simply caused by the small size of source and target in-domain corpora.

7.2 Future work

7.2.1 Larger in-domain corpora

A natural extension of this thesis would be to rerun the pipeline again, but this time gathering much larger in-domain corpora, presumably by scraping from multiple sources with more keywords. Hopefully, the amount of *recognized* words would increase.

7.2.2 DALI

Unfortunately, we did not achieve the same results as the DALI method, discussed in Section 3.4. Nonetheless, our work could still be expanded to mimic the DALI method more, for example using a SMT system to train a SMT model on the pseudo-parallel in-domain corpus, instead of the NMT system used in DALI.

7.2.3 Phrase-based evaluation with multiple phrase tables

Moses can utilize multiple phrase tables[1] during the learning and translation processes. A natural extension of the proposed pipeline would be to take the phrase table generated from in-domain data and combine it with a phrase table generated from a large parallel corpus, with the reasoning being the domain-specific words not present in the latter phrase table would be included in the former one. This would allow to evaluate the domain adaptation in a phrase context, utilizing traditional metrics such as *BLEU*[2].

An implementation of this process was attempted, but unfortunately the technical issues caused by excessive memory requirements (more than 500 GB of RAM was required) were not successfully resolved in time.

[1] http://www.statmt.org/moses/?n=Advanced.Modelsntoc7
[2] https://en.wikipedia.org/wiki/BLEU

Bibliography

[1] Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, August 2019. Association for Computational Linguistics.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[3] Jonáš Kratochvíl. Domain adaptation of automatic speech recognition for czech language, 2020.

[4] Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy, July 2019. Association for Computational Linguistics.

[5] Ivana Kvapilíková, Dominik Macháček, and Ondřej Bojar. Cuni systems for the unsupervised news translation task in wmt 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 241–248, Florence, Italy, August 2019. Association for Computational Linguistics.

[6] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019, 2018.

[7] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[8] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[9] 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 Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[10] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[11] Jorg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[12] Jana Straková, Milan Straka, and Jan Hajič. Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

List of Figures

List of Tables