Master Thesis

Domain adaptation of end-to-end ASR via n-gram language modelling

Tessel Wisman

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Sophie Arnoult
2^{nd} reader: Hennie van der Vliet

Submitted: June 30th, 2022

Abstract

This thesis concerns domain adaptation of automatic speech recognition (ASR) engines via n-gram language model adaptation. We formulate different strategies to adapt an end-to-end ASR engine trained on generic domain data towards two topic domains: biomedical sciences and politics. The strategies implemented are: building domain-specific n-gram language models, creating hybrid language models that are trained on generic- and in-domain data, and both Bayesian and linear interpolation between generic and domain-specific language models. Using Bayesian interpolation, we achieve a best result of 16.3% relative word error rate (WER) reduction with respect to the baseline for biomedical sciences and a 4.3% relative WER reduction for the politics domain. From this, we conclude that language model adaptation is a successful strategy to domain adaptation in ASR, optimized using Bayesian interpolation between generic- and domain models. This adaptation yields both a relative WER improvement as well as large successes in accurate key-term transcription and improved transcript readability, while avoiding over-adaptation and only losing up to 0.7% relative WER in generic domain performance.

Declaration of Authorship

I, Tessel Adinda Wisman, declare that this thesis, titled *Domain adaptation of end-to-end ASR via n-gram language modelling* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 30 June 2022

Signed: Tessel Wisman 

Acknowledgments

First of all, I would like to thank Nithin Holla for his excellent supervision during my internship. Without his gentle introduction to the world of automatic speech recognition, accurate advice and constant support this thesis would not have been possible.

Second, but not less important, I would like to thank my thesis supervisor Sophie Arnoult for guiding me through the writing process and pushing me into the right directions every time I needed it.

I would also like to thank Amberscript for giving me the opportunity to learn so much about ASR and the business world, and to the members of the science team and all other colleagues that have supported me during my internship.

Finally, I would like to thank all my friends for always being there with me in the library and helping me stay motivated. Special thanks goes out to Lydia, not only for her unconditional support throughout my academic career but also for always making time to accurately read, understand and review my academic writing.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

Automatic speech recognition (ASR) is the field in natural language processing that specialises in the conversion of speech (audio) to text. Where manual creation of subtitles and transcriptions is labour-intensive and costly, developments in ASR have made it possible to offer high-quality automated transcription and captioning of audio data with minimal or no human intervention, improving accessibility of the ever-increasing amounts of video and audio material available online. Ideally, ASR techniques can provide added understanding and value in many practical cases in the form of subtitling or transcript availability in e.g. political meetings, science lectures, health care etc. However, depending on the implementation, the performance of generic ASR engines may drop as models have to deal with noise, specific jargon or rare subject material. In such a circumstance, it is feasible to look into *domain adaptation*, the field in machine learning where we aim to adapt a model that works well for a certain source data distribution to a (related) target distribution.

In order to successfully convert speech to text, an ASR engine is in need of two components: an *acoustic model* that can learn the relation between the audio signal and some form of textual representation, and a *language model* that can learn the probabilities of words occurring in natural language, both individually as well as following each other in a sequence. There is a variety of acoustic signals that may map to the same word: there are millions of different human voices, a lot different dialects in each language and audio may be recorded under varying noisy circumstances. While this audio-to-word mapping may seem obvious to human ears, a computer will be in need of machine learning techniques in order to resolve such ambiguity. The language model component aids in disambiguating potential candidate words based on the likeliness that this particular word follows the words that were previously detected, based on its general knowledge of language.

For many years, ASR has relied on explicit modelling of phonetic features using toolkits such as *Kaldi* (Povey et al., 2011), *Sphinx-4* (Walker et al., 2004), *HTK* (Young et al., 2002) and *Julius* (Lee et al., 2001). In this conventional approach, there are several different steps to the acoustic component. This includes the process of extracting relevant features in the acoustic signal, implementing an acoustic model (usually a type of a hidden Markov model) that learns the relation between observed audio signals and the expected phonemes, and a lexical pronunciation model that maps these phoneme components to words (Ghai and Singh, 2012). While this approach used by all dominant toolkits has been successful, it requires explicit alignment of audio-to-phoneme and phoneme-to-word relationships. As a result, the creation and improvement of such models is reserved to linguistic and signal processing experts, while transcription is restricted to using phonemes explicitly coded in the pronunciation dictionary and language model.

More recently, new approaches using deep learning techniques have made the process of building ASR engines more accessible. These *end-to-end* ASR implementations yield promising results without the need for this explicit modelling, replacing part of the engineering process and simplifying ASR architecture significantly (Wang et al., 2019). End-to-end models directly map speech to text using a single neural network architecture that can automatically learn pronunciation and language information. A typical end-to-end speech recognition architecture consists of an encoder, that maps a speech signal to a feature sequence, an aligner, which aligns this feature sequence to language, and a decoder that maps the aligned features to text output. While conventional models need a pronunciation lexicon, which maps the phonemes recognized by the acoustic model to words and needs to be curated by linguists, an end-to-end model uses a lexicon only to improve spelling of the character-level acoustic model output. A pronunciation lexicon and language model are mandatory in the conventional pipeline, while an end-to-end model essentially learns these steps in one pass. An external lexicon and language model are not required, but improve results significantly and remain relevant in fine-tuning and improving the quality of transcription (Kannan et al., 2018).

End-to-end models do not require explicit feature engineering, but are data-driven and require vast amounts of data in order to learn speech-to-text mapping internally. Modern implementations such as wav2vec 2.0 (Baevski et al., 2020) show that pre-training the Transformer-based architecture on unlabelled audio data to acquire general knowledge of speech and then fine-tuning on transcribed speech-to-text corpora, can generate ASR pipelines that reach state-of-the-art performance.

1.0.1 Problem definition

Despite these large training corpora modern end-to-end models are built from, it is virtually impossible for a general model to cover all types of speech, themes and subjects. As mentioned earlier, the practical requirements for using ASR can differ in terms of acoustic circumstances, language use, syntax and specific jargon compared to a generic domain. The frequency of words may depend heavily on the current discussion topic: hearing words like *DNA* or *cytoplasm* in a biology class is not very surprising, while one does does not expect to hear the same terms in a business meeting. This not only goes for individual words, but also for the probability of sequences: it is likely that the word *prime* will be followed by *minister* when transcribing politics, while *prime suspect* is a more likely sequence in a crime series subtitle. Using one model trained on generic data can be considered naive and makes the language model prone to underestimating the probabilities of rare words and constructions, leading to erroneous transcriptions. For a specific use-case, it is therefore feasible to look at domain-specific ASR strategies, that enable building a model specifically designed to perform well with language in a certain (topic) domain.

An obvious approach for domain adaptation might be to rebuild the entire model from scratch from the acoustic component up to the language model for the specific domain in question. However, for large end-to-end data-driven ASR models, this would require a significant amount of domain-specific transcribed audio data, which is often not readily available or even hard to obtain. This makes a more efficient solution to domain adaptation desirable. Moreover, the question is whether acoustic modelling should even be considered a bottleneck in adapting models toward content domains: for a single language, the acoustic signals should be relatively similar. The component in ASR that actually stores most information about which words and syntactic construction occur is the language model. Reserving the fine-tuning and tailoring towards a specific domain for the language model can prove a simple yet effective way to perform topic-wise domain adaptation in automatic speech recognition.

The idea of using *language model adaptation* to adapt ASR engines has been previously explored in combination with conventional acoustic modelling by e.g. Nanjo and Kawahara (2003) and Hsu and Glass (2006); Hsu (2009), where authors were able to decrease word error rate (WER) on topic-specific lectures compared to using a generic model. These studies show that language model adaptation is can be an effective strategy for ASR domain adaptation in the conventional architecture. However, the question is if the effects of language model adaptation extend their usefulness in an end-to-end architecture. This research aims to provide an in-depth evaluation of the best strategies for language model adaptation in ASR, as well as a critical reflection on its added value, practical limitations and prerequisites.

1.0.2 Research question and outline

In this thesis, experiments are performed with different domain-specific n-gram language modelling strategies in the ASR pipeline with the objective to formulate an optimal strategy for domain adaptation. We aim to answer the following research question:

What is the impact of domain-specific n-gram language modelling on performance of end-to-end automated speech recognition engines?

Two different topic domains are investigated: *biomedical sciences* and *politics*. We examine the effect of replacing a standard generic language model with different language model adaptation strategies: domain-only models, that are trained on exclusively in-domain data, hybrid language models, that are trained on mixed corpora of domain- and generic data and (linear- and Bayesian) interpolated language models. These strategies are evaluated with respect to the following factors:

- The optimal strategy to compose/train language models to adapt to a domain looking at WER and the quality of the transcript.
- The selection of data for language modelling, taking into account data scarceness for difficult domains and making sure to cover both jargon-rich in-domain language as well as spontaneous speech.
- The influence of corpus size on language model quality, considering low-resource domains with limited data availability.

Each language model configuration is evaluated both individually as well as part of the ASR pipeline, reporting WER (word error rate) and relative improvement with respect to the baseline ASR engine. A relative WER improvement of 16.3% was achieved with the best language model adaptation in the biomedical domain; for politics, we report 4.3% WER improvement. Furthermore, the use of this strategy only causes 0.7% WER gain when transcribing generic audio, avoiding any overadaptation issues and keeping the model applicable in broad circumstances. These results show that domain adaptation of ASR can be achieved via n-gram language model adaptation, using a Bayesian interpolation strategy between language models built from in-domain data and a generic model.

The outline of this thesis is as follows: chapter 2 provides a background to ASR and relevant concepts, followed by chapter 3 where we discuss the research setup, data and language modelling experiments. Chapter 4 presents the results and an error analysis by comparing the baseline engine with an engine where the proposed improvements are implemented, followed

by a discussion in chapter 5. Finally, we conclude our insights and recommendations in chapter6.

Chapter 2

Literature review and related work

Automatic speech recognition models are designed to recognise and transcribe (speech) audio to text. The following section contains a brief review of conventional- and end-to-end architectures, the language model component and related work in domain adaptation.

2.1 Acoustic models in ASR

2.1.1 Conventional methods

As briefly touched upon in the introduction, a typical conventional ASR pipeline relies on carefully engineered steps. This includes a feature extractor for signal parameterisation, a Gaussian mixture model (GMM) for acoustic scoring, a hidden Markov model for sequence modelling and a decoder module that generates an output hypothesis incorporating the acoustic scores, language model and pronunciation model. We will cover the architecture of an ASR pipeline in the following paragraphs.

As described by Aggarwal and Dave (2011) in their comprehensive review of acoustic modelling in ASR, the goal of speech recognition can be statistically formulated as finding a word sequence $W = \{w_1, w_2, ...w_n\}$ given the acoustic vector observed for speech $X = \{x_1, x_2, ..., x_T\}$, estimating the most likely word sequence

$$\hat{W} = \arg\max_{W} P(W|X) \tag{2.1}$$

In order to go from the raw speech signal to some sort of textual representation, we first need to extract features from the acoustic signal. Various signal processing methods can be used to generate parametric spectral representations, such as mel-frequency cepstral coefficient (MFCC), perceptual linear prediction (PLP), temporal patterns (TRAPs) and wavelets. These spectral representations are then processed by pattern classification algorithms in the acoustic component to create a mapping from the speech signal features to basic speech units (phones or phonemes). Algorithms that have been implemented for this purpose are Bayesian networks, neural networks, support vector machines and most predominantly hidden Markov models (Aggarwal and Dave, 2011).

The acoustic model generates a sequence of (phone) symbols that are most likely to correspond to the speech signal. This sequence of most likely phonemes however does not yet correspond to the final text output. Depending on the acoustic circumstances, phones may be transcribed wrongly, creating a sequence of most likely phonemes that does not correspond to a word in natural language.

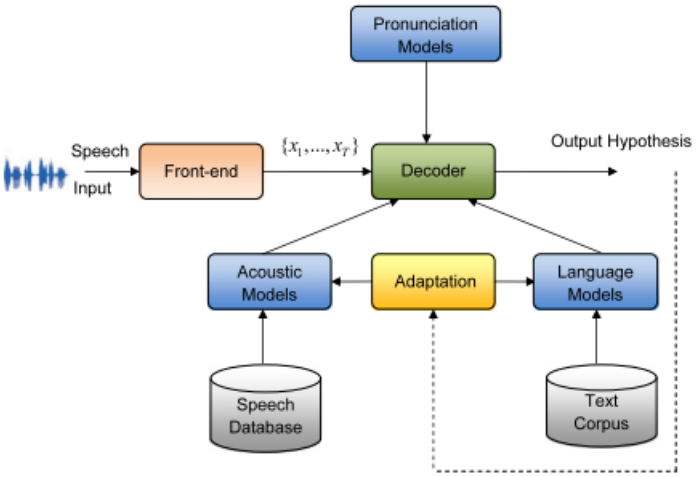

Figure 2.1: *A block diagram of the ASR pipeline*
from: Aggarwal and Dave (2011)

To ensure that the transcribed word both actually exists in the language and makes sense in the language context, the sequence of phonemes is verified by looking it up in the *pronunciation dictionary*. The pronunciation dictionary, or lexicon, contains a list of all words that are known by the ASR model and which combinations of phonemes correspond to them. A probabilistic inference between the list of most likely phonemes generated by the acoustic model and the pronunciation dictionary is then made to obtain the final most likely transcript of the speech signal. A conventional ASR model can only recognize speech units that it has learned during training, but would be able to compose a new word that was not in the training data if it consists of speech units that the model has seen, provided that the entire word itself is available in the pronunciation dictionary.

One more module is at play in the ASR pipeline: the *language model*. In this component, the general probability of words occurring in the language is encoded, based on the likeliness of the word itself as well as the context that usually surrounds the words. More about language models follows in section 2.2.

The final transcription is delivered by the decoder module, that searches for a correct word sequence using the output of the acoustic model, the language model and the pronunciation lexicon. The most widely used approach for this is Viterbi beam search, that searches the most likely sequence of words based on the path of tokens with the highest probability.

2.1.2 End-to-end models

As can be seen in figure 2.1, a conventional ASR architecture requires several separate components in order to provide transcription: the acoustic model, pronunciation model and language model are all key input for the decoder module. *End-to-end systems* reduce the complexity of ASR systems compared to the conventional approach by directly mapping the input signal (speech) to text via a neural architecture. With this direct mapping, the process of feature extraction of acoustic signals is performed internally by the neural network, and there is no longer need for explicit alignment to phonemes and words or a closed look-up dictionary. End-to-end models directly map a sequence of acoustic input into a sequence of on words, sub- words or characters and graphemes using a neural encoder-decoder architecture (Wang et al., 2019).

The key idea of encoder-decoder architecture is that an input sequence $X = \{x_1, ..., x_T\}$ is converted to an output sequence $L = \{l_1, ..., l_N\}$. This is achieved through two steps in the two components: first creating a mapping (*encoding*) of X to a hidden feature representation $F = \{f_1, ..., f_T\}$, while simultaneously calculating another mapping (*decoding*) from F to L. Comparing to the conventional approach to ASR as described in the previous section, the end-to-end model effectively combines feature extraction, acoustic modelling and phoneme-to-character mapping in one algorithm.

One of the more complicated processes in conventional acoustic modelling is that while training, acoustic signals and their features have to be explicitly aligned (*hard alignment*) to their corresponding phonemes. End-to-end ASR is capable of *soft alignment*, where each audio frame corresponds to all possible transcriptions with a certain probability distribution. The optimal probability mapping from speech signal unit to corresponding speech unit can then be learned by the acoustic model.

There are multiple approaches to soft alignment, one of which is *connectionist temporal classification (CTC)* (Graves et al., 2006). CTC enumerates all possible hard alignments and then aggregates them in order to achieve soft alignment. Given the input sequence $X = \{x_1, ..., x_T\}$ of length T where we want to find the most likely word in the vocabulary that corresponds, this essentially means that there is no hard requirement for this word to have the same length T as the input sequence (Wang et al., 2019).

Looking back at the encoder-decoder architecture where we try to map from X to output L, we do not decode directly from F to L. Instead, each f_t at time step t from feature vector $F = \{f_1, ..., f_T\}$ is converted to a probability distribution sequence y_t, composing vector $Y = \{y_1, ..., y_T\}$. Each y_t indicates how likely f_t corresponds to a any possible word in the vocabulary. The most likely *path* π from X to L given the input vector, $p(\pi|X)$, can be seen as the product of probabilities in Y for each character in π at each time step t:

$$p(\pi|X) = \prod_{t=1}^{T} y_t^{\pi_t} \tag{2.2}$$

under the constraint that the most likely word predicted by π is in the vocabulary.

As can be seen, this most likely mapping from input to output still speaks of a fixed number of time steps t. Soft alignment, the possibility to deviate from this, is achieved by mostly two processes:

1. Merging the same contiguous labels: if multiple input timesteps x_t and x_{t+1} map towards the same character, they are merged into one output label.

2. Deleting blank labels: if an x_t corresponds to no label, it is deleted.

With an architecture as described above, it is possible to go from speech signal to a set of most likely characters in one pass. Merging the different modules from conventional ASR into one model has two main advantages: joint training of the multiple ASR components allows the algorithm to configure a global optimum instead of having to optimize the modules one-by-one, and reduces the processing complexity. However, as is clear from the representation of $p(\pi|X)$ in equation 2.2, CTC makes an important assumption in modelling path probability as a product of individual time steps: that each mapping from x_t to π_t is independent. When seeing each character in a word as independent, a CTC-based model is consequently unable to do *language modelling*; the likeliness of the characters is not evaluated in context, making it purely an acoustic model. In order to create words that actually make sense in context, both

lexicon- and language model scoring is incorporated in the CTC process. A lexicon in end-to-end ASR is similar to a pronunciation dictionary, but contains a mapping from character spelling to words (instead of phonemes to words), making sure the output words exist in the language. For both the lexicon and the language model, each character's probability in context is interpolated with its initial probability from the acoustic model.

An implementation of the above end-to-end type model is used in this research project, which is wav2vec 2.0 (Baevski et al., 2020). A further explanation of this model architecture follows in chapter 3.

2.2 Language modelling

With either of the acoustic models in section 2.1, the character- or word-level output is subsequently scored by the language model. Language models are a key component in ASR systems, searching for the most likely word given the context in the language. The language model score is incorporated with a certain weight, reinforcing or discouraging the word formed by the acoustic model output. A language model takes the several word hypotheses generated by the acoustic models and scores them with a probability for each sequence of words (Aggarwal and Dave, 2011). Scores of the acoustic model and language model are then linearly interpolated in the decoding step to maximize:

$$y* = \arg\max_{y} \{\log P_{ASR}(y|x) + \beta \log P_{LM}(y)\} \tag{2.3}$$

for acoustic input x and word output y, where β is a parameter that defines the importance of the external language model (Inaguma et al., 2019). There are essentially two types of language models: statistical models, that compute probabilities of words occurring in a sentence, and neural network models. N-gram models are used in this research and will be discussed below.

2.2.1 N-gram models

The most commonly known and widely used implementation of language modelling in ASR is the *n-gram language model*. An n-gram language model is formally known as 'a probability distribution $p(s)$ over strings s, reflecting how frequently a string s occurs in a sentence' (Chen and Goodman, 1999). N-gram language models compute the probability of sequences of words under the assumption that the probability of word w_i occurring depends directly on the probability of the sequence of previous words in the string.

This means we can predict the probability of a word occurring now, based on how likely it was to occur after the words we have already seen in the past. However, looking at the entire history of a word is too complex: language is creative, and the chances that there is no knowledge available about the entire previous context is high. Therefore, n-gram language models approximate the history of previous words by only looking at $n-1$ previous words: $w_{i-1}...w_{i-n}$ for an n-gram model. A unigram model, for which $n = 1$, stores for each word the probability that this word occurs in the corpus by itself. A bigram model ($n = 2$) reflects the probability of this word given the previous word. This can be extended to trigrams, fourgrams, fivegrams and further to obtain more and more insight into how language in the corpus is structured. The higher the n-gram order, the better the corpus is captured in its probability distribution, but the lower the chance that the model can predict an unseen corpus just as well. The probability of a string s consisting of k words w_i in an n-gram language model is formally represented as:

$$\prod_{i=1}^{k} p(w_i|w_{i-n+1}^{i-1}) \tag{2.4}$$

where w_{i-n+1}^{i-1} is the n-gram word history.

Smoothing

The conceptual simplicity of n-gram language models comes with a few complications. Despite the fact that we already approximate a word's context by only looking at n previous words, the chances that a finite corpus will not contain all valid n-gram sequences in the current language are high, even for large training corpora. As a consequence, some n-gram sequences that occur outside the training corpus will have zero probabilities even though they are valid sequences. Because the probability of the string is computed as the product of word probabilities, this will lead to the entire string having zero probability: something that is definitely to be avoided.

In order to prevent this, *smoothing techniques* are implemented, which make sure these zero-probabilities are converted to a non-zero value. As discussed in Chen and Goodman (1999) referring to Lidstone (1920), Johnson (1932) and Jeffreys (1998), a simplest solution to this issue is *additive smoothing*: add δ (typically $0 < \delta < 1$) to the count of each n-gram. Although this prevents zero counts, this smoothing method adds little informative value for unseen n-grams. Many more elaborate techniques have been designed over the years. The Good-Turing Estimate (Good, 1953) is a central measure in many smoothing techniques, assuming that for any *n*-gram that occurs r times, we estimate its count looking at the number of other n-grams that also occur r times. Techniques like Jelinek-Mercer smoothing (Jelinek, 1980) use another core idea: for higher order n-grams that don't have an estimate, we just rely on the lower-order $n-1$ probability. A linearly *interpolated* estimate for an n-gram model then becomes a weighted mixture between the *n*-gram probability and the interpolated *(n-1)*-gram probability:

$$P_I(w_i|w_{i-n+1}^{i-1}) = \lambda P(w_i|w_{i-n+1}^{i-1}) + (1-\lambda)P_I(w_i|w_{i-n+2}^{i-1}) \tag{2.5}$$

with fixed weights λ. Other well-known methods building on these concepts are Katz Smoothing (Katz, 1987), Witten-Bell smoothing (Witten and Bell, 1991; Bell et al., 1990), absolute discounting (Ney et al., 1994) and Kneser-Ney smoothing (Kneser and Ney, 1995). Kneser-Ney smoothing will be used as smoothing technique in language model building: see 3.8.2.

2.2.2 Neural language models

Another class of language models is *neural language models* (NLMs), which have proved to outperform n-gram language models on several benchmarks, showing relative WER reduction of several percentage points (Raju et al., 2019; Jozefowicz et al., 2016). Such recurrent NLMs and LSTM-based implementations are generally better at capturing a language, considering they take into account unlimited context history; while n-gram models only look at $n-1$ previous words, NLMs learn a distributed representation of each word as well as the probability function for the word sequence in context (Bengio et al., 2000).

This unlimited context history leads to an exponential explosion in decoder search space, as the number of hypotheses that arises from the acoustical model can be very large (Raju et al., 2019). This is also the main reason that *n-gram* language models are still most widely implemented: the improved performance of neural models comes with added computational complexity, which slows down the already complicated ASR pipeline. Instead of using the

NLM directly in ASR decoding, a common approach is to run the decoding in two passes, using a simple n-gram language model to narrow down the search space and rescoring the output with the stronger NLM in the second pass. While narrowing down the number of hypotheses with an n-gram model in the first pass reduces computational complexity for the NLM, adding a second component still slows down the pipeline. Furthermore, the strength of the NLM is not maximally exploited anymore: if the n-gram language model already throws away good hypotheses, the NLM will not consider them either.

2.3 Domain adaptation strategies in ASR

Domain adaptation in ASR can refer to different subtasks such as adapting models to work for different types of speech within a language ((Hwang et al., 2022), as well adapting speech models to low-resource languages (Anoop et al., 2021) or clean to noisy speech (Manohar et al., 2018). These approaches fine-tune end-to-end acoustic models on domain data in combination with updated language models. Techniques for domain adaptation include data augmentation, multi-task learning, teacher-student models or semi-supervised training (Wotherspoon et al., 2021). These techniques however approach the problem of domain adaptation to be solved with at least some proportion of acoustic data, creating a model that can transcribe domain-specific audio by learning from the source domain and fine-tuning on the target domain.

Using *only* the language model for domain adaptation in speech recognition is a lower-resource approach, suitable in cases where the need for domain adaptation does not come from acoustic difficulty, but differences in the topic- or subject matter. *Language model adaptation*, the procedure of changing a language model to better fit a target corpus, has been reviewed by Bellegarda (2001). In language model adaptation, we search to maintain an adequate representation of our target data under changing conditions. In our case, this change in condition is the change of domain, where we want to compensate for any mismatch between training and speech recognition.

The key idea of language model adaptation is generally centered around language model *interpolation*, given a domain-specific (smaller) corpus A and a background (large) corpus B. According to Bellegarda (2001), there are three main approaches to language model adaptation. The first is simple model *interpolation* where a language model from A is combined with a model from B according to preset weighting parameters. A more elaborate approach is determining the optimal model mixture using *constraint specification*, where the domain-specific corpus A is used to extract features that the adapted language model should satisfy. A third approach is using *metadata extraction*, where corpus A is used to extract topic-, syntactic- or semantic information about the subject matter to determine the ideal model adaptation weighting.

The question how language models can be useful for adaptation to different topics in ASR has been explored by e.g. Nanjo and Kawahara (2003) as well as Hsu and Glass (2006). The authors examine the application of hidden Markov models with Latent Dirichlet Allocation as a way to do topic modelling of language model training corpora, in order to create interpolated topic-matching n-gram models models. In their work, the authors address the need for target-domain specific language modelling in ASR, specifically regarding the challenge of fitting text corpora to audio domains.

Following this line of work, Hsu (2009) proposes a new interpolation technique for n-gram language models to solve the challenge of academic lecture transcription. In academic lectures, the presence of dense scientific subject material goes hand in hand with a high de-

gree of speaker spontaneity (hesitations, repetitions, rephrasing). This proposes a challenge in accurate language model training, as corpora that cover scientific terms and concepts will inevitably not cover this spontaneous speech, while general speech corpora will miss out on special terminologies and the topics covered in the lecture. Language model adaptation can also be performed with neural language models, which has been a central focus in more recent work such as Raju et al. (2019); Liu et al. (2021).

Chapter 3

Methodology

The following chapter outlines the methods used for our domain adaptation research. The characteristics of our ASR setup and task definition are first covered in section 3.1, followed by the evaluation metrics in 3.2. In order to investigate the effects of building domain-specific ASR engines with custom language models, we then need to select relevant domains and create a framework to test the transcription quality for these domains in section 3.3. A collection of language model training data was gathered (section 3.4) as well as an evaluation set, which has been curated to test performance over domains (section 3.5). Different language modelling strategies are designed in order to address potential shortcomings of the baseline model in section 3.7, from which we derive our experimental setup as discussed in 3.8.

3.1 Model characteristics and task definition

3.1.1 Acoustic model

The acoustic model evaluated throughout the entire project is the Amberscript end-to-end English engine. Amberscript is an Amsterdam-based company that specializes in speech-to-text, using ASR to provide automatic- and improved captioning and transcripts. This model is an implementation of a semi-supervised, pre-trained model similar to wav2vec 2.0 (Baevski et al., 2020). Wav2vec is a model from Facebook AI that can be trained on a large corpus using self-supervised learning, reducing the amount of labelled data that is needed by using transcribed data only for fine-tuning. The general data representations are learned in an unsupervised way, which means the model needs only audio data (without transcriptions) for the pre-training process.

The model takes audio data as input and encodes this using a multi-layer convolutional network. When passed through the encoder-decoder architecture, it produces speech representations for each (pre-defined) timeframe of length t in the audio signal. The model learns to convert these speech representations to text by applying techniques derived from *masked language modelling*: randomly masking a proportion of the speech representations in an audio signal, and learning to distinguish the true content of the masked proportion from a set of distractors. Before performing the contrastive task of distinguishing the true latent from distractors, the masked vectors (latent speech representations) are transformed into discrete signals using a quantization module and fed to a Transformer context network, obtaining contextualised representations.

After the model has learned its basic representations of speech by this unsupervised masked language modelling technique, it still has to be fine-tuned. This is done by adding a randomly

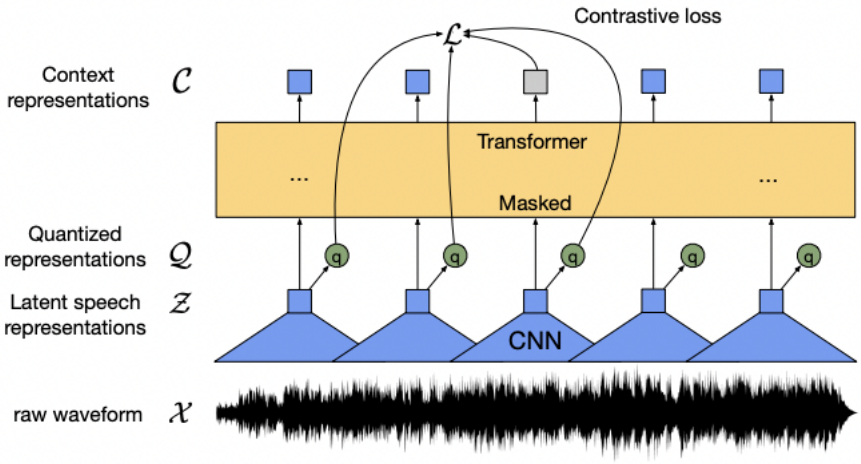

Figure 3.1: *Illustration of the Baevski et al. (2020) framework*
The waveform representations X are mapped to latent representations Z from which the model jointly learns contextualized speech representations C and an inventory of discretized speech units Q.

initialised linear projection on top of the context network that projects to C classes, representing the character vocabulary. The most likely transcript to an acoustic signal is then obtained by minimising CTC loss over these classes. The results are incorporated with a language model using beam search (either n-gram or Transformer-based) to obtain final speech-to-text.

3.1.2 Language models

All language models implemented in this project are n-gram models, following Amberscript's setup and practical considerations. Our n-gram models are built using *Kneser-Ney smoothing* (Kneser and Ney, 1995), which is an optimized version of absolute discounting (Ney et al., 1994). Absolute discounting builds on the principle of interpolating between a higher-order and a lower-order n-gram distribution. For a certain n-gram count, we subtract a fixed discount factor δ (Chen and Goodman, 1999). If this distracted value is larger than zero, its probability is obtained by dividing it by the total number of n-gram counts for this word; else, we take zero probability. This score is interpolated with the n-gram probability of the one-lower order n-gram, describing the final n-gram probability of word w_i given n-gram history w_{i-n+1}^{i-1} as:

$$p_{abs}(w_i|w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^{i}) - \delta, 0)}{\sum_{w_i} c(w_{i-n+1}^{i})} + \lambda(w_{i-n+1}^{i}) p_{abs}(w_i|w_{i-n+2}^{i-1}) \tag{3.1}$$

with normalization constant λ:

$$\lambda(w_{i-n+1}^{i}) = \frac{\delta}{c(w_{i-n+1}^{i-1})} |w_i : c(w_{i-n+1}^{i} w_i) > 0| \tag{3.2}$$

The problem with absolute discounting is that when a higher-order n-gram probability is low, relying heavily on a lower n-gram probability is not always a good idea. The classic example is the phrase *San Francisco*: in a text where this bigram occurs a lot, the unigram probability of *Francisco* will be high. In a sentence where higher-order n-gram probability is low and the

model reverts to unigram probability, *Francisco* will then be considered a likely word to follow any other word, while in reality being very unlikely as a unigram without *San* preceding it.

Kneser-Ney smoothing addresses this problem by adapting unigram probabilities to not only represent the number of occurrences of the word, but making it proportional to the number of different words that it can follow. This means instead of just using $p_{abs}(w_{i-n+2}^{i})$, we use the *continuation probability* of a word, proportional to the number of higher-order n-grams it completes. For a bigram model, this is the probability that word w_{i-1} is followed by w_i:

$$P_{cont}(w_i) = \frac{|w_{i-1} : c(w_{i-1}w_i) > 0|}{\sum_j |w_{i-1} : c(w_{i-1}w_i) > 0|} \tag{3.3}$$

which stands for the number of n-grams that the final word can complete, divided by the total number of possible n-grams. In the case of *San Francisco*, *Francisco* can only complete one n-gram, so its continuation probability will be low.

This makes the generalized formula for Kneser-Ney smoothing:

$$p_{KN}(w_i|w_{i-n+1}^{i-1}) = \frac{\max(c(w_{i-n+1}^{i}) - \delta, 0)}{\sum_{w_i} c(w_{i-n+1}^{i})} + \lambda(w_{i-n+1}^{i})\frac{|w_{i-n+1} : c(w_{i-n+1}^{i}) > 0|}{\sum_i |w_{i-n+1} : c(w_{i-n+1}^{i-1}) > 0|} \tag{3.4}$$

3.1.3 Task overview

The purpose of this research is to adapt the ASR pipeline towards specific domains by incorporating a customized n-gram language model. In end-to-end ASR, the output of the acoustic model is a character-level sequence, which is scored by a language model in order to produce natural language in context.

As explained in the introduction, the context-level is expected to be most characteristic to a topic domain, which is why domain adaptation is approached at the language model level. This means the setup of this project is as follows:

1. The baseline setup of the ASR is evaluated on two specific topic-domains.
2. Experiments with several language model configurations are performed, with the goal of improving the generic setup.
3. Language model performance is evaluated individually by computing *perplexity* on a validation set.
4. Language models are evaluated as a component in the ASR pipeline focussing on *word error rate gain* and *transcript quality* improvement on both the domain datasets as well as on the generic dataset (to account for over-adaptation).

A more in-depth discussion of this setup and related concepts follows in the sections below.

3.2 Evaluation metrics

In order to evaluate the language models, their contribution to the ASR and the influence in domain adaptation, we look at different metrics and aspects of evaluation. Initially, quantitative performance is measured in terms of *perplexity* and *word error rate*. This section introduces these evaluation metrics as well as *transcript quality*, another point of interest.

3.2.1 Perplexity

Perplexity will be used in *intrinsic evaluation* to measure the quality of our language models. Perplexity can be defined as the normalized inverse probability of the test set and is given by the following formula:

$$PP(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, ..., w_N})} \tag{3.5}$$

where $P(w_1, w_2, ..., w_N)$ indicates the probability that the word sequences found in the test set follow from the sequence probability computed in the training corpus. The lower the perplexity, the less the model is 'surprised' by the evaluation dataset, thus the better the language model reflects the domain.

3.2.2 Word error rate

A common metric to measure the performance of speech recognition engines is word error rate (WER). Word error rate is a simple measure that divides the number of errors by the number of words in the reference transcript and will be used as our *extrinsic* evaluation measure. The number of errors is determined by the sum of *deletions (DEL)*, the number of *insertions (INS)* and *substitutions (SUB)*, generating

$$WER = \frac{SUB + DEL + INS}{N} \tag{3.6}$$

for N words in a corpus.

Additionally to WER, we also report *relative word error rate difference*: the relative improvement of WER with respect to the baseline in percent. As baseline WER may differ across domains, this gives us more insights in the relative impact of adapting language models.

3.2.3 Transcript quality

While WER is effective in its simplicity, the number of errors does not always reflect transcript quality as a whole. Examining the rate of substitutions, deletions and insertions on their own is therefore informative as well. For example, a high deletion rate that is caused by a noisy environment that makes words indetectable for the ASR is problematic in any circumstance, while deletions may have a different impact in other cases. When transcribing stuttering speakers that repeat words and interject many *uhm*'s, *like*'s and *yeah*'s, a higher deletion rate might actually improve a transcripts readability and quality. Similarly, the occasional insertion of an interjection does not make an automatically generated caption significantly more difficult to understand, while giving a completely unrelated substitution of a key concept term does.

A multi-applicable and practical automated speech recognition system should not only strive to obtain the lowest possible WER and focus on the quantity of correctly transcribed words, but also take into account a notion of *transcript quality*. While this is a vague concept, carefully analysing the errors an engine made and their impact on readability should be given accurate attention in ASR research. In our analysis, we will therefore discuss not only WER, but also the impact of changes in transcripts with respect to the following points:

- The impact of deletions, substitutions and insertions on the overall clarity of the transcript: can the meaning of the sentence be understood or inferred?

- The amount and impact of changes that should be made to obtain a perfect transcript, as would be important in an implementation where ASR for example assists a manual subtitler/transcriber.

3.3 Selecting domains

To test the hypothesis that domain specific language models can improve ASR performance compared to generic language models, we first need to select relevant domains. Our baseline, the *generic domain* is defined as a mixed collection of speech that covers different subjects but is not specifically limited to certain topics or speech circumstances. A generic language model is then trained on a mixture of speech data, adapted to cover for all possible speech.

Preferably, domain adaptation experiments are performed on both a domain that is expected to be hard to cover for with a generic language model, as well as a domain that should pose less challenges, in order to be able to effectively estimate the impact of customising language models. The selected domains for this purpose are *biomedical science education* and *politics*.

Biomedical science is a scientific field that contains a lot of specific jargon that is not always known by the common speaker: think of proteines, diseases and anatomical terms. This jargon is likely only partially or not at all covered by a generic language model. This means we can expect low probability for these words to be selected in speech recognition. The hypothesis is therefore that ASR should show a relatively large improvement when experimenting with domain-specific adapations of the language model, making it our *high-effect* domain. In order to reflect a potential practical use of biomedical speech recognition, we select the academic lectures subdomain.

A second domain should still benefit from domain adaptation. However, it is interesting to select a domain where the influence of domain adaptation is expected to be at a different level in order to infer where and when this strategy is worth to implement. Political speech is expected to be less 'difficult' for the generic model to handle; while jargon will be present, politicians are discuss a broad range of themes and actualities that can be thought of as generic to some degree. As a consequence, politics will therefore be our *lower-effect* domain.

3.4 Training corpora

To adapt language models towards the selected domain, biomedical- and politics text corpora were collected. The text data and used partitions are discussed in the section below.

3.4.1 Biomedical domain

In search for a large, open-source corpus that contains biomedical jargon, text data for biomedical language model training was scraped from Wikipedia. Making sure that only relevant biomedical texts are included, scraping was performed by selecting specific categories and collecting the contents of all pages in these categories. The categories included in the language model are displayed in table 3.1. Equations and headers were removed from the raw page contents. The total corpus size of the scraped Wikipedia articles makes up 5.7 million words, which has been divided into a train (Bio-TRAIN) set of 4.8 million words, and a tuning (Bio-TUN) set of 0.9 million words that will be used to determine the ideal weights in language model adaptation (see section 3.7).

Anatomy	Human anatomy	Human biology
Physiology	Organs (anatomy)	Tissues (biology)
Human physiology	Medical terminology	Cell biology
Membrane biology	Cell anatomy	Cells
Cell signalling	Embryology	Cellular respiration
Metabolism	Cellular processes	Molecular biology
Protein structure	Biochemisty	Proteins
Biomolecules	Peptides	Genetics
Cytogenetics	Molecular genetics	Medical genetics
Population genetics	Gene expression	Neuroscience
Molecular neuroscience	Brain anatomy	Neurochemistry
Brain	Nervous system	Neuroanatomy
Cellular neuroscience	Immunology	Molecular oncology
Oncology		

Table 3.1: *Wikipedia categories included in the training corpus of the biomedical domain*

3.4.2 Political domain

In the politics domain, two separate corpora were used for different experiments: a *British parliament* dataset and the *European Parliament Proceedings Parallel Corpus* (Europarl). Both corpora are described in the paragraphs below.

British Parliament corpus

As our political domain test set is focused on British politics, the closer a language model can approach British political speech, the better. The main corpus for political language model training consists of transcripts of meetings of the British House of Commons[1]. Transcripts were scraped from 01-01-2021 up to 29-03-2022, obtaining a dataset of approximately 15 million words. As a preprocessing step, the frequent abbreviation *hon.* was converted back to *honourable* (from the phrase *right honourable friend*, a common way politicians refer to each other in Parliament). In order to be able to compare the language model to the biomedical domain model, the final corpus was brought down to 5.7 million words and a train-validation split was created resulting in a 4.8 million word training corpus (Poli-TRAIN) and 0.9 million word tuning/validation set (Poli-TUN/VAL). As we want to minimize the risk of creating a language model that is largely biased towards only current topics discussed in parliament, coherent speech paragraphs were randomly sampled from the 15 million words corpus over a larger time span opposed to only including more recent transcripts.

Europarl corpus

Since 5.7 million words compose a relatively small language model, a second external corpus is added for the politics domain in order to be able to investigate the effect of model size. The English partition of the *Europarl* corpus (PoliEU-TRAIN) (Koehn, 2005) is therefore included in one of the language model experiments for politics. This dataset is a 55 million word collection of European Parliament proceedings.

[1] Available under an *Open Parliament License v3.0*, scraped from *https://www.theyworkforyou.com/api*.

3.4.3 Generic domain

During experimentation, partitions of generic data are also used in language model training in order to potentially improve the domain-specific model implementations. To make the models containing generic data comparable to the language modelling baseline, the data that was used to train Amberscript's generic language model was recollected. This corpus consists of a 639 million word composition of open-source datasets extended with company data, and contains a wide range of spoken- and written language sources. A collection of randomly sample sentences from these corpora is used in language model building throughout this project (Gen-TRAIN).

3.5 Extrinsic evaluation data

In order to test the effect of designing new language models, audio data has been gathered for each domain to evaluate the ASR engine performance. For the biomedical domain, the model is evaluated on a collection of educational- and lecture video's available on YouTube that consider topics like DNA, the immune system, neurobiology and cell biology. For the politics domain, performance is evaluated on a collection of video excerpts of British Parliamentary events[2]. This includes debates in the House of Commons, the House of Lords and meetings of the Treasury Committee and Children and Family Act 2014 Committee.

Both domain-specific datasets are curated to contain $\pm$4h of speech and include a variety of speaker accents and genders. In the biomedical domain, an effort was made to include different varieties of English such as American (en-US), British (en-GB), South African (en-ZA) and Australian (en-AU), simulating an education environment that features teachers and experts from different backgrounds. The politics dataset however does limit to British (en-GB) and Irish (en-IE) including several regional dialects, in order to simulate a practical use case of a domain model as political practices are often country-specific. A more in-depth description of the evaluation dataset is available in appendix A (7).

In order to be able to compute performance statistics, ground-truth transcriptions are needed for all evaluation data audio. These were obtained via Amberscript's online platform, where manual transcribers created *verbatim* (true to utterance) transcriptions for each file that are used as reference for computing error statistics.

Finally, a generic audio dataset has been composed to evaluate potential over-adaptation of language model implementations to a specific domain. This dataset consists of two hours of Amberscript company data. A complete overview of the training corpora and evaluation data used can be found in table 3.2.

3.6 Intrinsic evaluation data

To evaluate the language models prior to running the entire ASR pipeline, validation sets were constructed for both domains. For the politics domain, the language models are validated on Poli-VAL, the 0.9 million word partition of the scraped data from 3.4.2. This validation set consists of transcripts similar to the language model training data, which is expected to be a good representation of the evaluation data. For the biomedical domain on the other hand, we anticipate a larger gap between training and extrinsic evaluation data subdomains. To better

[2] Available for download on *parliamentlive.tv*.

reflect the extrinsic evaluation set, the validation set is selected to be a 1 hour transcript of a biology lecture (Bio-VAL).

Dataset	Domain	Specification	Size
Bio-TRAIN	Biomedical	Wikipedia corpus	4.8M
Poli-TRAIN	Politics	Parliament transcript	4.8M
PoliEU-TRAIN	Politics	Europarl corpus	60M
Gen-TRAIN	Generic	Amberscript sample	4.8M
Bio-VAL	Biomedical	Lecture transcript	11.1K
Bio-TUN	Biomedical	Wikipedia corpus	0.9M
Poli-VAL/TUN	Politics	Parliament transcript	0.9M
Gen-TUN	Generic	Amberscript sample	0.9M
Bio-TEST	Biomedical	Lecture audio	4hrs
Poli-TEST	Politics	Parliament debates	4hrs

Table 3.2: *Overview of datasets used and their purpose*

3.7 Domain adaptation

Using the datasets described in previous section 3.4, language models were created according to three main different strategies. The goal of this research is to compare these strategies and gain insight not only in the general effect of domain adaptation via language models, but also to find the optimal approach. Three methods of training language models are discussed in this section: implementing *domain-specific-only* models, combining the domain-specific and general training corpora to obtain a *hybrid* model, and language model adaptation by *interpolating* between the generic and domain specific model. Another point of interest is the effect of corpus size and the trade-off between domain match and size.

3.7.1 Baseline

In order to compare the language models that are generated with the above domain-specific corpora and determine an optimal strategy, a *baseline* was established for the entire evaluation set. The baseline was obtained by running the ASR pipeline while incorporating the default generic language model. This language model has been trained on the complete generic domain corpus from Amberscript (639 million words) and calculated using KenLM (Heafield, 2011) with n-gram order 4 and pruning threshold 1. For this generic baseline model we expect suboptimal performance on domain-specific terms, as the probabilities drawn from a large, mixed corpus will likely underestimate both biomedical jargon as well as words specific to British politics.

3.7.2 Domain-only language models

A first strategy to domain adaptation is integrating language models into the ASR pipeline that are exclusively trained on the collected domain-specific training corpora. Such a model will be specifically tuned to the training material and the domain. The success of this approach depends largely on how well the language model training data is expected to cover the audio to be transcribed: if their subdomains are very close, this should work well; if the training data is no perfect match, performance may drop.

For the biomedical domain we are using Wikipedia data as training corpus, while our evaluation data consists of lectures. This means we can expect a partial mismatch between the language model data and the evaluation data that might influence the performance of a domain-only language model integration negatively. For the political domain, this effect is not necessarily expected: our corpus consist of Parliament transcripts which are very similar to the test audio.

Lexicon

In addition to the language model, the *lexicon* also plays an important role in the ASR's linguistic component. The lexicon acts as a constraint of what words are possible in the language; for end-to-end ASR, a word not being present in the lexicon significantly reduces the chances of producing it in a transcript. When performing domain adaptation, it is also important to take this component into account and expand the lexicon with domain-specific words if the model is to transcribe these words correctly.

There are two possible strategies for the lexicon when adapting the language model. The first one is to rebuild the lexicon with domain-only data, which should work adequately if the training corpus is large enough and covers the test domain. In practical cases however, when performing domain adaptation from an existing generic ASR engine, it makes more sense to simply *extend* the lexicon: since it is only a constraint, adapting fully to the domain language model means an unnecessary loss of generality. With domain specific language models, we experiment with both building the lexicon and language model both with domain-only data, as well extending the generic lexicon from the 639 million word language model used in the baseline with the domain vocabulary. All other strategies discussed below will use the latter strategy.

3.7.3 Hybrid language models

While using a domain-only language model may work well in cases where the audio to transcribe exactly matches the domain the language model was trained on, there is also a risk of overfitting to the text corpus. This can be especially problematic in ASR if we think about an important language style difference that persists even if the training corpus matches the domain topic very well: written language opposed to spoken language. Word use and especially word sequence in spoken language is different compared to written language; the former contains more incomplete sentences, stutters and repetitions and is less formal. While finding domain-specific corpora that contain the relevant jargon might be challenging already, the chances of finding suitable domain text data that also matches spontaneous speech are even slimmer.

Enriching the domain-specific corpus with a portion of generic data decreases the risk of the final model missing out on generic words as well as spontaneous speech, given that our generic corpus consists for a large part of speech transcripts. For both domains, a language model is trained on a balanced combination of generic and in-domain data, accompanied by a combined lexicon. For the biomedical domain, where the language model training corpus is more scientific in nature and of a narrow domain, partial integration of the generic corpus is expected to achieve better results than using domain-only training data. For the political domain, where our corpus consists of speech transcripts, the integration of the generic data is expected to have less impact.

3.7.4 Language model interpolation

Instead of training on a joint corpus, *interpolation* is a widely used technique to combine two language models. In interpolation, language models are combined by drawing probabilities from separate language models trained on separate corpora (Pusateri et al., 2019). Compared to joint training, interpolation provides the opportunity to customize the weights given to either domain. Furthermore, it allows for combining two already existing language models, dismissing the need for complete retraining in each new data configuration.

The probability of a word w given an n-gram history h can then be expressed as the sum of probabilities over all different domains i as

$$p(w|h) = \sum_i p(w, i|h) = \sum_i p(i|h)p(w|i, h) \tag{3.7}$$

In this research, we will use two different methods for interpolating between domains: *linear interpolation* and *Bayesian interpolation*.

Linear interpolation

By plugging Bayes' rule:

$$p(i|h) = \frac{p(i)p(h|i)}{p(h)} \tag{3.8}$$

into equation 3.1, we can derive the interpolated estimate of the probability of a word given its history, $p^{int}(w|h)$, as the sum over all word histories for each individual domain multiplied by the probability of this history existing in the current domain, divided by the probability of the word history occurring over all domains:

$$p^{int}(w|h) = \sum_i \frac{\lambda_i p(h|i)}{p(h)} p_i(w|h) \tag{3.9}$$

Here, constant $\lambda_i = p(i)$ represents the prior probability of the current domain i. In linear interpolation, the strong assumption is then made that word history h is independent of the domain i: $p(h|i) = p(h)$. This simplifies equation 3.3 to

$$p^{LI}(w|h) = \sum_i \lambda_i p_i(w|h) \tag{3.10}$$

By doing this, we essentially assume that both domains share the same vocabulary, which is a naive assumption given our domain-adaptation premise.

The interpolation weight λ_i for each domain can be either explicitly specified or learned by tuning the interpolated model on a representative text file, optimizing the parameters using gradient descent.

Bayesian interpolation

The second approach to interpolation implemented is Bayesian interpolation. This method does not assume that word history is independent of the domain, which is expected to show a significant improvement with respect to linear interpolation if the interpolated models are not very similar. Since this is the case for both our domains (especially biomedical sciences), Bayesian interpolation is expected to be more successful than linear interpolation.

Bayesian interpolation does not disregard a relation between word history and domain and computes $p(h|i)$ as the probability of word sequence h given by the ith component language

model. This is denoted $p_i(h)$ (Pusateri et al., 2019). Plugging this estimate into Equation 3.3, the formula of Bayesian interpolation is:

$$p^{BI}(w|h) = \sum_i \frac{\lambda_i p_i(h) p_i(w|h)}{\sum_j \lambda_j p_j(h)} \tag{3.11}$$

3.8 Experimental setup

The following section discusses the final experimental setup, as derived from the domain adaptation strategies discussed above.

3.8.1 Configurations

The different language modelling experiments are conducted as listed below. Each experiment is assigned an acronym, which will be used as a reference in the remainder of this paper. Where not explicitly mentioned, the lexicon used consists of the generic baseline lexicon extended with the domain vocabulary from the training data. An overview can be found in table 3.3.

ACRONYM	Corpora used	LM strategy	Specifics
BASE	639M word generic	Baseline	-
GEN-S	Gen-TRAIN	-	-
DOM-dLX	TRAIN	-	Domain lexicon
DOM-cLX	TRAIN	-	Combined lexicon
HYBRID-EU	(Poli-)TRAIN + EUTRAIN	Hybrid	-
HYBRID-L	TRAIN + Gen-TRAIN	Hybrid	-
HYBRID-S	$\frac{1}{2}$ TRAIN + $\frac{1}{2}$ Gen-TRAIN	Hybrid	-
LINT-D	TRAIN + Gen-TRAIN	Linear int.	Tuned on TUN + Gen-TUN
LINT-V	TRAIN + Gen-TRAIN	Linear int.	VAL-tuned
BAYI-11	TRAIN + Gen-TRAIN	Bayesian int.	1:1 weights
BAYI-32	TRAIN + Gen-TRAIN	Bayesian int.	3:2 weights

Table 3.3: *Overview of the experiments from section 3.5 and their acronyms. Where not specified, a model was trained on both the Bio- and Poli-variant of the corpus.*

Generic domain experiments

Two experiments are performed that focus on only the generic domain as training data source:

- **BASE**: the baseline experiment, setting the starting point for potential improvement by language model changes.
- **GEN-S**: since the baseline model is particularly large (639 million words) compared to the domain models used in the rest of the experiments (4.8 million words), there is a substantial chance that the domain adaptation strategies in this research will display relatively weak performance due to this size difference. To make sure such effects can be detected, an 'equal match' was created by training a language model on in order to obtain more insight in relative improvements as well as the importance of language model size.

Domain model experiments

The following experiments are performed with domain-only corpora:

- **DOM-dLX**: a language model trained on exclusively domain data for both domains (Bio- or Poli-TRAIN), accompanied by a lexicon from the same corpus.
- **DOM-cLX**: the same domain language model, accompanied by a lexicon that contains the generic vocabulary extended with the domain vocabulary.

Hybrid model experiments

- **HYBRID-EU**: as the domain specific models are both quite small, we look at the use of a larger domain language model for politics only, trained on PoliEU-TRAIN and Poli-TRAIN together obtaining a total corpus size of 60 million words. While the Europarl corpus consists of politics transcripts (from the EU parliament), its subdomain is further from our British politics test set than our small British language model. A point of interest is therefore also the tradeoff between exact domain match and size. Due to limited access to data in the biomedical domain, a similar experiment was not repeated for biomedical sciences.
- **HYBRID-L**: a hybrid language model trained on both Bio- or Poli-TRAIN and with a combined lexicon.
- **HYBRID-S**: another model that should give more insight in the effect of model size is this hybrid language model trained on *half* of Bio- or Poli-TRAIN and *half* of Gen-TRAIN.

Interpolation experiments

- **LINT-D**: a model that is linearly interpolated between a domain model as used in DOM-experiments and the small generic model GEN-S. The model is tuned with gradient descent on Bio- or Poli-TUN and Gen-TUN, which leads to an approximately 1:1 weight ratio (see table 3.4 for the tuning weights).
- **LINT-V**: a model that is linearly interpolated between a domain model as used in DOM-experiments and the small generic model GEN-S. The model is tuned with gradient descent on Bio- or Poli-VAL. This means our intrinsic evaluation of this model will be biased and should be interpreted carefully, but the overlap between tuning- and validation data does not influence the extrinsic evaluation. Adapting the language model more directly towards the needs of the domain is therefore still an interesting experimental strategy (again, see table 3.4 for the tuning weights). For biomedical sciences, this means a slight leaning towards the domain model, while the generic model influence is almost eliminated for politics. As the politics validation set is very similar to the language model training data, it is no surprise the tuning leads to favouring the domain language model.
- **BAYI-11**: a Bayesian interpolated model between the Bio- or Poli-DOM model and GEN-S model with a weight ratio 1:1, approximating the weighting from LINT-D.
- **BAYI-32**: a Bayesian interpolated model between Bio- or Poli-DOM and GEN-S with a weight ration 3:2, giving some dominance to the domain model probabilities approximating the weighting from LINT-V. The slight overweighting of the domain is expected to give better results than using the 1:1 ratio.

Model	Generic weight	Domain weight
Biomedical		
LINT-D	0.52	0.49
LINT-V	0.43	0.58
Politics		
LINT-D	0.58	0.45
LINT-V	0.09	0.94

Table 3.4: *Interpolation weights*

3.8.2 Specifications and hyperparameters

Language models were trained using KenLM (Heafield, 2011) except for Bayesian interpolation models for which OpenGRM (Roark et al., 2012) is used. All models use n-gram order 4, keeping language model complexity similar to the generic reference language model used as baseline. This decision was validated experimentally (see 4.2.1). No pruning constant was used as all language models are relatively small.

Chapter 4

Results

This chapter discusses the evaluation of the customized language models. First, an extensive analysis of the baseline results is provided in section 4.1. Second, a perplexity evaluation of all language models on the validation set is provided in section 4.2. Subsequently, we discuss the WER statistics of the different language model pipelines in section 4.3. An evaluation of (over-)adaptation of the models on a generic testset can be found in section 4.4.

4.1 Baseline

In order to determine the optimal domain adaptation strategy and desired language model capabilities, a baseline (BASE) was established for the entire evaluation set. We will cover the mistakes and shortcomings of the baseline model in the following section to illustrate the need for domain-specific language modelling.

4.1.1 Intrinsic evaluation

The perplexity scores of the generic language model on both the intrinsic validation set as well as the (transcripts of the) evaluation set are displayed in table 4.1. The perplexity of the model is significantly higher on the biomedical domain, which along the expected lines considering domain distance. The difference between validation- and test perplexity is lower for the biomedical domain (29.1) than for politics (42.3), indicating that the biomedical validation set is a better reflection of the test set than the political validation set. Another interesting observation is that the test set perplexity of the generic model on the generic domain is higher (375.1) than on the politics domain (318.4). This means that the generic model is a surprisingly good predictor for the politics domain.

Domain	**Val. perplexity**	**Test perplexity**
Generic	-	375.1
Politics	276.1	318.4
Biomedical	528.4	557.5

Table 4.1: *Perplexity of the generic language model on validation and test set*

Domain		WER	SUB	DEL	INS
Biomedical	**Average**	8.6	4.2	3.7	0.8
	Standard Deviation	2.4	1.0	2.5	0.6
Politics	**Average**	13.9	4.9	7.9	1.1
	Standard Deviation	3.2	0.8	3.0	0.4

Table 4.2: *Baseline performance of the ASR with the generic Amberscript language model*

4.1.2 Extrinsic evaluation

Generally, the WER baseline for both domains is already quite solid with an average of 8.6 WER for the biomedical domain dataset and 13.9 WER for the politics domain dataset (see table 4.2). In the following section, we look further into the baseline results.

Biomedical domain

An average word error rate of 8.6 is quite low in the jargon-rich biomedical domain and exceeds initial expectations. This overall good performance can probably be attributed to the quality of the dataset; the files are low-noise and generally clear-spoken. The interesting part therefore lies in the types of errors made. The hypothesis for the biomedical domain was that the terms that are specific to biomedical sciences and are likely to occur frequently in our domain-specific dataset, are not extensively covered by the generic language model.

For a lot of domain-specific words, the model is correct in some occasions, while failing at the exact same words in other contexts. We can suspect that the generic language model includes some biomedical training data, but the probability of jargon words is not large enough to encourage the ASR to transcribe these words in a variety of situations. For example, the word *eukaryotes* is substituted by *carriers, years, chariots* or *you Karioti*, *prokaryotes* becomes *precarious* and *DNA* is substituted by *Diana, day* or *diner*. The model also does not recognize proteins and sugars as *cytosine (citizens)* and *guanine (gunn and)* and *ribose (boswell)*. An example sentence can be seen in figure 4.1. Words in red represent ASR mistakes; corrected words of interested are marked in bold.

Generic engine transcription:
"The first group of organisms are called **carriers**. The second group of organisms are called **precarious**."

True transcript:
"The first group of organisms are called **eukaryotes**. The second group of organisms are called **prokaryotes**."

Figure 4.1: *Example sentences where the baseline engine provides incorrect transcription of domain-specific jargon*

Political domain

Contrary to what might be expected looking at assumed domain distance, the WER for the politics domain is higher than for the biomedical domain with an average of 13.7. However, when we look into the statistics, more things seem to be at play than simply missing

out on Parliament-specific terms; a substantial amount of errors are related to either the acoustic model or post-processing. The deletion rate is relatively high with 7.6 compared to 3.6 in the biomedical domain while the number of substitutions is similar (4.9 for politics versus 4.2 for biomedical). A parliamentary debate contains more interruptions, stutters and crosstalk than biomedical lectures. These are not always picked up and lead to a higher number of deletions compared to the reference file. Another cause for higher WER is spelling: the baseline model produces American spelling, while the reference transcripts are British. While this issue is not a point of interest in this research, it is important to keep its potential influence on WER in mind.

Other than that, we see a similar type of substitutions as in the biomedical domain where terms that are very specific to the dataset are substituted by more generic words: *Tory MP's* becomes *tops*, *Prime Minister* is transcribed as *port minister* and the term *right honourable friend* (a common way colleagues refer to each other in Parliament) is transcribed as *right one friend*. These are terms that ought to be given more priority when a domain-specific language model is implemented. In debates or meetings about specific topics, substitutions are also common: *Ukrainians* is transcribed as *uranian*, and *surrogacy* and *fertility* become *sagas* and *fatality*. Again, an example can be found in figure 4.2.

Generic engine transcription:
"Well actually, I **have** the first **quest hold bate** on the subject in 2015".

True transcript:
"Well actually, I **held** the first **Westminster Hall debate** on this subject in 2015."

Figure 4.2: *Example sentences where the baseline engine provides incorrect transcription of domain-specific jargon*

4.1.3 Summary

Summarizing the findings in the baseline experiments, the generic end-to-end model has quite a good performance on both domains. This good performance can be partially attributed to the fact that the evaluation data contains little noise, which gives the acoustic model better chances of transcribing the right content to begin with. Updating the language model with (a large proportion of domain-specific) text is expected to contribute and improve it further by increasing the probabilities of domain-specific terms and including new words. We can update our hypotheses as follows:

- We expect the biomedical domain to benefit relatively more from domain-specific language modelling than the political domain, as the former contains more domain-specific terms. The political domain is expected to show relative improvement, but is limited as the error rate is also largely caused by the number of deletions which is within the range of the acoustic model.

- In analysing domain adaptation, the focus should be on the relative improvement and mostly on changes in the amount of substitutions the model makes.

- Spelling errors appear to play a role in the higher word error rate for the political domain. Using a British corpus for language modelling might help increase the probabilities given to British spelling compared to American spelling. This should be taken into account in case of significant changes in WER.

4.2 Domain adaptation: intrinsic evaluation

After establishing the baseline obtained by the use of a generic language model, the different language mode strategies from 3.7 are evaluated. A primary intrinsic evaluation of the language models on their own can be found in the section below. Next, the language models are evaluated as component of the ASR pipeline in section 4.3.

4.2.1 N-gram order

As a first evaluation, *perplexity* is computed for all language models on the validation set. The decision to use n-gram order 4 was validated by computing perplexity for bigrams, trigrams, fourgrams and fivegrams in two basic configurations: domain-specific language models and models trained on a combination of domain- and generic data. A visualization of perplexity vs. n-gram order can be found in figure 4.3, accompanied by table 4.3. All demonstrated models follow the same pattern: perplexity significantly decreases when moving from bigrams to trigrams; fourgrams still exhibit relative improvement to trigrams but this stagnation of perplexities comes more or less to a stop for fivegram models. This validates our decision to take fourgram models as our general implementation.

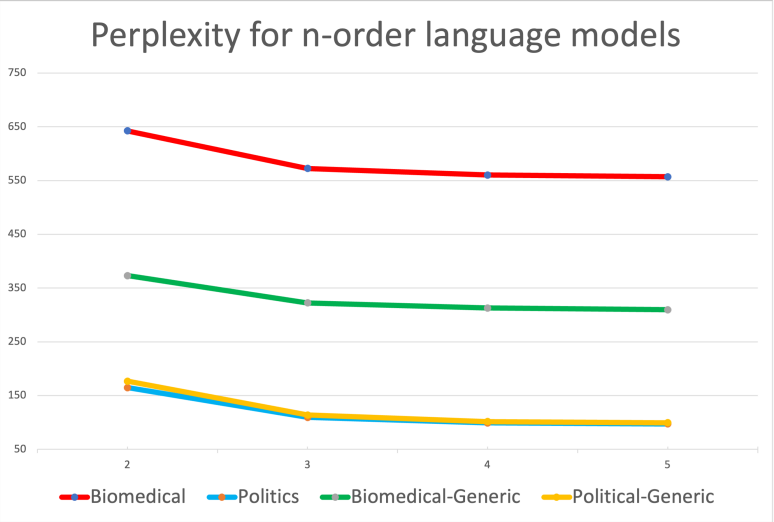

Figure 4.3: *Perplexity plotted for different n-gram orders*

4.2.2 Language model strategies

Biomedical

Table 4.4 displays perplexity of all generated language models on the validation set.

For the biomedical domain, the best model is trained on a combined corpus of domain- and generic data with a perplexity score of **312.9** and a reduction of **40.1%**. Similar to the baseline

n-gram order	Biomedical	Bio-Generic	Political	Politic-Generic
2	642.5	372.9	164.7	176.7
3	572.4	322.2	109.4	113.7
4	560.3	312.9	99.1	101.6
5	556.9	309.6	97.2	99.3

Table 4.3: *Perplexity for different n-gram orders.*

Language Model	Perplexity	Difference	Rel. difference
Biomedical			
BASE	528.4	-	-
GEN-S	715.4	+187	+26.2%
DOM	560.3	+31.9	+6.0%
HYBRID-L	**312.9**	-215.5	-40.1%
HYBRID-S	345.0	-183.4	-34.7 %
LINT-D	476.6	-51.8	-9.8%
LINT-V *	468.0	-60.4	-11.4&
BAYI-11	397.86	-130.5	-24.7%
BAYI-32	397.36	-131.0	-24.8%
Politics			
BASE	276.1	-	-
GEN-S	313.17	+37.07	+13.4%
DOM	**99.1**	-177	-64.1%
HYBRID-EU	110.08	-166.0	-60.1%
HYBRID-L	101.59	-174.5	-63.2%
HYBRID-S	119.7	-156.4	-56.6%
LINT-D	116.6	-159.5	-57.8%
LINT-V *	117.6	-158.5	-57.4%
BAYI-11	139.25	-136.9	-49.7%
BAYI-32	136.76	-139.3	-50.5%

Table 4.4: *Perplexity scores of the language models*
DOM refers to the language model used in either DOM-dLX or DOM-cLX, as the lexicon has no influence in perplexity.

** These perplexity scores have been verified on the same data the model is tuned on, which means this result should be interpreted carefully.*

results, modelling of the biomedical domain appears to be harder than the political domain: the perplexity scores are significantly higher overall. As hypothesized, the baseline-exceeding perplexity score of 560 by DOM indicates that the validation data shows too few overlap with the language model training data domain to make a domain model viable on its own. Biomedical lectures are a tricky combination of spontaneous speech and jargon; considering this, a 6% perplexity loss compared to the baseline for the domain model is not surprising. This is also likely the reason that incorporating generic data is a more successful strategy in language modelling, making HYBRID-L the best model perplexity-wise, followed with a notable but not surprising difference of 32.1 points by its smaller counterpart HYBRID-S. The LINT-models show the least improvement with respect to the baseline, but LINT-V outperforms LINT-D.

Politics

For the politics domain, we can see that the best model appears to be the domain-specific model with a very good perplexity score of **99.1** and a relative gain compared to the baseline of **64.1%**. As the validation set of the politics domain is from the same source as the language model training data (House of Commons transcripts), it is no surprise that this model shows the most promising results. All other language model configurations show relative gain compared to the baseline language model, except for the small generic language model, which is also to be expected. Linear interpolation shows better results than Bayesian interpolation in both configurations. The LINT-D model tuned on a combination of domain- and generic data performs slightly better than LINT-V, tuned on validation data only.

An interesting fact is that Bayesian interpolation appears to do a better job at modelling our validation data for biomedical sciences, while linear interpolation takes the lead with politics. This difference may be caused by the difference in domain distance: since politics is closer to the generic domain (as was confirmed by the baseline perplexity in table 4.1), the naive assumption of linear interpolation that word history is independent of the domain causes less issues than for the biomedical domain, where domain distance is larger.

Following our expectations, the language model that is trained on half of both generic and domain data (HYBRID-S) is significantly less successful than the model where all data was used HYBRID-L with scores of 119.7 versus 101.59. On the other hand, the even larger HYBRID-EU model performs worse than HYBRID-L. Again, this may be caused by the fact that the validation data is an exact domain match to the politics data used in all other models and appears to be covered very well by the generic language model. As HYBRID-EU is trained on European Parliament transcripts, it might be disadvantaged having neither a dominant influence of British Parliament data nor any generic data incorporated. However, the fact that this model is significantly larger might prove more important in our evaluation of the test audio in the ASR pipeline. If more types of political meetings than only the House of Commons are to be transcribed and if there is more variation in topics, the larger vocabulary and size of HYBRID-EU might compensate for where it lacks in subdomain-specificness.

4.2.3 Summary

Summarizing our findings above, we add the following expectations regarding the extrinsic language model evaluation component:

- Domain-only language modelling has some potential in the politics domain, while for biomedical sciences some influence of generic data is definitely needed to obtain any performance increase.
- Building on the perplexity scores, the hybrid language models appear to have an advantage on the interpolated models.
- Linear interpolation shows better perplexity scores for politics, while Bayesian interpolation takes the lead in biomedical sciences. Because the politics domain is closer to the generic domain, the history independence assumption has limited influence on interpolation results.

4.3 Doman adaptation: extrinsic evaluation

In the following section, we discuss the final results of running the ASR pipeline with all proposed language model strategies incorporated. Strategies are discussed by domain. An

overview of all WER statistics for all models is given in table 4.5 and visualized in figure 4.4.

Experiment	WER	Rel. WER diff.	SUB	DEL	INS
Biomedical					
BASE	8.6	-	4.2	3.7	0.8
GEN-S	9.3	+8.1%	4.8	3.9	0.6
DOM-dLX	10.9	+26.7%	3.7	6.6	0.5
DOM-cLX	8.6	0%	3.6	4.6	0. 5
HYBRID-L	7.3	-15.1%	3.2	3.5	0.5
HYBRID-S	7.4	-14%	3.3	3.5	0.5
LINT-D	8.4	-2.3%	3.9	4	0.5
LINT-V	8.2	-4.7%	3.7	4	0.5
BAYI-11	7.2	-16.3%	3.2	3.5	0.5
BAYI-32	**7.2**	**-16.3%**	**3.1**	**3.5**	**0.5**
Politics					
BASE	13.9	-	4.9	7.9	1.1
GEN-S	14.4	+3.6%	5.1	8.2	1.1
DOM-dLX	15.4	+10.8&	4.6	9.8	1.0
DOM-cLX	14.0	+0.7%	4.6	8.4	1.0
HYBRID-EU	14.2	+2.2%	4.6	8.6	1
HYBRID-L	13.5	-2.9%	4.5	7.9	1.1
HYBRID-S	13.6	-2.2%	4.5	8.0	1.1
LINT-D	13.9	0%	4.7	8.1	1.1
LINT-V	14.1	+1.4%	4.7	8.4	1
BAYI-11	**13.3**	**-4.3%**	**4.4**	**7.8**	**1.1**
BAYI-32	13.4	-3.6 %	4.4	7.9	1.1

Table 4.5: *Averaged word error rate statistics of running the ASR pipeline with different language models*

4.3.1 Biomedical science domain

Balanced generic language model

The difference between the baseline generic model and the small generic model is significant, with the latter showing a 8.1% WER loss over the biomedical test set. Reducing the generic model size increases the (if not sporadic) presence of in-domain training data that is already present in the generic corpus. However, considering the enormous size difference (640 million words vs. 4.8 million words), the effect is not that severe.

Domain-only language models

The language model that was trained on only domain-specific data shows an increase to 10.9 WER if the lexicon is also limited to only corpus vocabulary (DOM-dLX) and a similar WER of 8.6 to the baseline with a combined lexicon (DOM-dLX). This corresponds to the higher perplexity score that we have seen for the domain-only language model and the fact that the

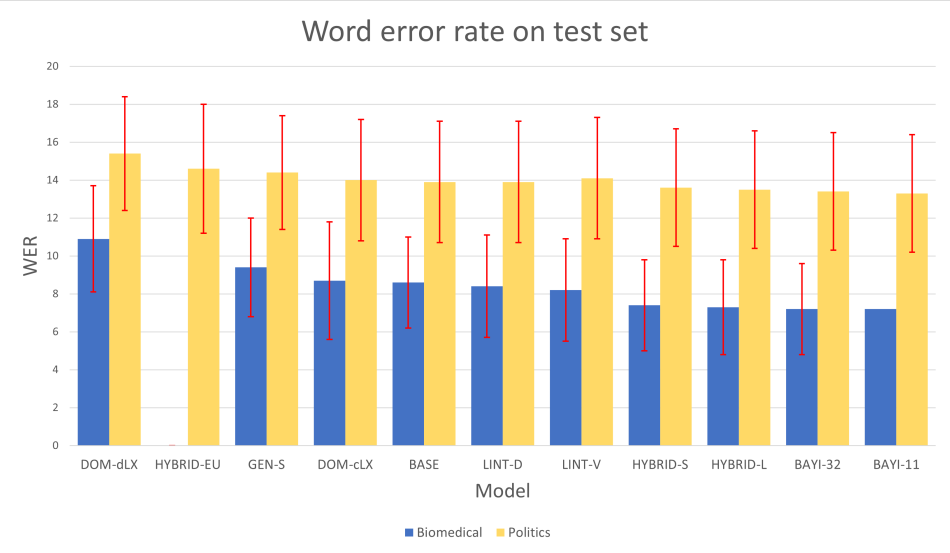

Figure 4.4: *Plot of word error rates over all models*
The models are sorted by their performance averaged over both domains (decreasing WER); the red line represents the standard deviation.

biomedical lectures from the test set contain a combination of domain-specific terminology and spontaneous speech, which the Wikipedia corpus does not completely cover.

Nevertheless, we can see that the number of substitutions (SUB) *does* go down for both these model configurations: from 4.2 for BASE to 3.7 for DOM-dLX and even 3.6 DOM-dLX if we use the combined lexicon. Deletions (DEL) on the other hand increase from 3.7 to 6.6 and 4.6 respectively. This indicates that while the model may lose generality, it might be more capable of transcribing specific biomedical terms. On the other hand, when the language model lacks coverage of spontaneous expressions the model appears to be more eager to delete these.

We examine an example transcript for the DOM-dLX model in 4.5. There are quite a few deletions in this transcript, which are all frequent, non-content words like *a*, *the* and *I*. Meanwhile domain words *eukaryotic* and *DNA* are transcribed correctly. The reverse is true for the baseline transcript: here the accompanying words are correct, while the domain-jargon is substituted. Which of these deletions/wrong substitutions is worse, is up for discussion. When evaluating not only word error rate, but also transcript quality and the added value such ASR may have in automated subtitling or serving as a base for manual transcribers, it might be more effective in the end to accept the filler word deletions and at least get the content words right. More about this follows in the discussion in chapter 5.

Hybrid language models

As soon as some generic data is incorporated in the language model, performance improves with 14% for HYBRID-S and even 15% for HYBRID-L compared to the baseline. In terms of size effects, the double-sized domain-generic model HYBRID-L appears to have little advantage over the smaller HYBRID-S model. The distincition between these models is visible in a 0.1 point difference for substitutions yielding a 1.1% relative WER improvement for doubling corpus size.

Training a language model on this combination of biomedical jargon from the Wikipedia corpus and generic spoken data appears to decrease the amount of both substitutions as well

True transcript:
"The defining feature of a eukaryotic cell is the presence of a nucleus, which contains the DNA that controls the cell's activities."
Generic engine transcription:
"The defining feature of a **Critic** cell is the presence of a nucleus which contains the **diner** that controls the cell's activities."
Bio DOM-dLX transcription:
"The defining feature of **[DEL]** **eukaryotic** cell is the presence of **[DEL]** nucleus, which contains the **DNA** that controls the cell's activities."

Figure 4.5: *Example sentences illustrating the changes in deletions and substitutions when using Bio-DOM-dLX*

as deletions. An illustration can be observed in figure 4.6: while Bio DOM-dLX omits the *I*'s, HYBRID-L fixes this mistake.

Contrary to what might be expected from the perplexity scores in 4.2, HYBRID-L is not the best performing model when implemented in the ASR pipeline and loses to the Bayesian interpolation methods by 0.1-0.2 points. The differences between interpolation and joined corpus language model training will also be covered in the discussion.

DOM-cLX transcript:
[DEL] want to ask you one question before **[DEL]** move on.
True / HYBRID-L transcript:
I want to ask you one question before **I** move on.

Figure 4.6: *Example sentence illustrating the impact of generic training data in the language model*

Interpolated language models

With a small difference of 0.1 WER point to the domain-generic language model, the overall best biomedical model is the Bayesian interpolated implementation with a 3:2 domain-generic weight ratio (BAYI-32), achieving **7.2** WER. This model has also the lowest substitution ration (3.1); while weighting difference of BAYI-11 and BAYI-32 does not appear to have a large impact, the latter has a 0.1 point advantage in substitutions. An example transcript from BAYI-32 is provided in figure 4.7: here we see how the language model fixes the most important substitutions and deletions made by the baseline generic engine.

True transcript:
"Yeah. I got rid of the DNA. This is the mRNA that we were able to transcribe from that DNA."
Generic engine transcription:
"**Now** I got rid of the **[DEL]**. This is the **man** that we were able to transcribe from that **[DEL]**."
Bio-BAYI-32 transcription:
"**Now** I got rid of the **DNA**. This is the **mRNA** that we were able to transcribe from that **[DEL]**."

Figure 4.7: *Example sentences illustrating the changes in deletions and substitutions when using Bio BAYI-32*

The Bayesian interpolation method is significantly better than the linear interpolated models LINT-V (8.2 WER) and LINT-D (8.4 WER). As previously mentioned in the perplexity

validation 4.2, linear interpolation assumes that word history is independent of domain. This assumption is probably too naive given the large difference between the biomedical and generic domain. Still, corresponding with the expectations from perplexity scores, tuning on the validation set is more effective than tuning on a combination of domain and generic data.

4.3.2 Politics domain

The overall impact of language model adaptation in the politics domain is lower compared to the biomedical domain. The best performing model, BAYI-11, only loses 0.6 WER compared to the baseline (13.3 and 13.9 respectively), making it a relative WER decrease of **4.3%**. This is a far less significant effect compared to 16.3% decrease in the biomedical domain.

Balanced generic model

The small generic model GEN-S again shows lower performance than BASE (14.4 vs. 13.9), although the performance drop is less significant than for the biomedical domain. When further examining the differences between BASE and GEN-S, we can see that the performance drop for both domains is most importantly caused by an increase of substitutions. This increase is significantly lower in the politics domain, where it goes up by only 0.2 points, compared to 0.6 for the biomedical domain. As previously mentioned, it can be hypothesized that as the politics domain lies closer to the generic domain, the small generic model does a better job at covering for its vocabulary than for the biomedical domain.

Domain-only language model

Contrary to what one might expect based on the perplexity scores, the domain-only language model is no longer the best performing in the politics domain when incorporated in the ASR pipeline. It is even the worst performer with a WER of 15.4. The number of deletions increases significantly from 7.9 to 9.8 if we also only use an in-domain lexicon (DOM-dLX), and still goes up to 8.4 if the generic lexicon is kept (DOM-dLX). Despite that the politics training corpus consists of transcripts that are an exact domain match to our test set, these transcripts are cleaned, excluding repetitions and stuttering. Evaluating on similar clear transcript in the validation round has not posed a problem, but this may be different when applying the language model in noisy speech conditions. We can suspect that the discrepancy between train- and test data might have been underestimated and generic speech data is still needed to account for spontaneous, spoken language.

The type of improvements domain models make in the politics domain are actually quite similar to their contribution in the biomedical domain. The substitution rate improves with a modest 0.3 rate from 4.9 (DOM-dLX) to 4.6 (DOM-dLX); on the other hand deletions rise from 7.9 to 9.8 and 8.4. An example sentence is found in 4.8.

Hybrid language models

Similar to the biomedical domain, WER starts improving as soon as some generic data is incorporated. We see a relative improvement of 2.9% for the HYBRID-L model and 2.2% for the HYBRID-S model. Considering corpus size, doubling the size of the domain and generic corpus language model in HYBRID-L compared to HYBRID-S again does not have a large impact. Together with what we have seen for the biomedical corpus, this confirms the effectiveness of the language model adaptation on its own: sheer corpus size is not the most important factor.

True transcript:
"Mr. Speaker, at the very same time, Tory MPs were gathering across the street for a champagne bash in the Park Plaza."
Generic engine transcription:
"**[DEL]** At the very same time, **tops, we're** gathering across the street for a champagne bash in the park, **fabrication problems.**"
Politics DOM-dLX transcription:
"**[DEL]** At the very same time, **Tory MPS** were gathering across the street for **[DEL]** champagne **base** in the park **fabric of problems**."

Figure 4.8: *Example sentences illustrating the changes in deletions and substitutions when using Poli-DOM-dLX*

The HYBRID-EU model was hypothesized to provide a larger domain coverage for the politics domain. Its performance however is disappointing, not only as we have seen in perplexity score but also in WER performance. Word error rate goes up with 2.2% from the baseline to 14.2. Even more interesting, the HYBRID-EU model for the politics domain performs worse than the DOM-dLX model, even though being over 30 times larger. The problem is again mostly in the higher deletion rate: similar to the other domain-only language models, the HYBRID-EU model seems overeager to delete filler- and speech words such as *a*, *I* and *the*. This causes the deletion rate to increase from 7.9 to 9. The DOM-dLX model still shows a slightly lower deletion rate with 8.6.

In terms of substitutions, the HYBRID-EU model misses the probability mass to transcribe terms like the before mentioned *Tory* or *right honourable friend* correctly in all circumstances, while the model trained on exclusively British politics gets these terms in most of the cases. From this we can draw a careful conclusion that exact domain match in a smaller language model is to be chosen above a larger, medium domain-match language model.

Interpolated language models

A Bayesian interpolation between the generic and domain language model is again the most successful model with a **13.3** WER and a relative WER loss of **4.3%** for BAYI-11. The best strategy here is using 1:1 interpolation weights compared to 3:2 in the biomedical domain, making a 0.1 difference in the deletion rate.

An example sentence for this best language model configuration is seen in 4.9. We can see how the model improves on some substitutions while continuing to suffer from a high amount of deletions. While *SNP business speaker* is somewhat closer to the true transcript *SNP spokesperson* than *Boston Custom speaker*, the entire phrase *"Thank you Mr. Speaker"* is deleted by both the politics language model and the baseline model. While *Bridgend* is transcribed wrongly by the generic language model to *James*, the politics language model deletes it entirely. A significant proportion of the higher deletion rate in the politics domain is likely also caused by failure of the acoustic model; as these test files contain a lot of cross-talk and noise and deletions are present regardless of language model configuration.

While linear interpolation looked more promising in our perplexity validation (see 4.2), performance does not show any improvement with LINT-D and even appears to cause a rise in deletion rate with model LINT-V (7.9 to 8.4). Again, like previously mentioned, the discrepancy between the in-domain politics data and generic data is perhaps larger than expected, giving linear interpolation the same disadvantages as for the biomedical domain.

True transcript:
"We now come to SNP spokesperson Kerstin Boswell. Thank you Mr. Speaker, I'd like to add my best wishes to the honourable member for Bridgend."
Generic engine transcription:
"We now come to **Boston Custom speaker [DEL]. an** I'd like to add my best wishes to the honourable member for **James**."
Poli-BAYI-11 transcription:
"We now come to SNP **business speaker [DEL]**. I'd like to add my best wishes to the honourable member for **[DEL]**."

Figure 4.9: *Example sentences illustrating the changes in deletions and substitutions when using Poli-BAYI-11*

4.4 Over-adaptation

To measure how significantly the language models influence domain adaptation, all models were also tested on a generic evaluation set to visualise performance loss across the generic domain. These results are displayed in table 4.6.

Experiment	**WER**	**Rel. WER. diff.**	**SUB**	**DEL**	**INS**
BASE	15.3	-	-	-	-
GEN-S	15.5	+1.3%	7.8	6.6	1.1
Biomedical					
DOM-dLX	19.2	+20.3%	8.5	9.8	0.9
DOM-cLX	18.1	+15.5%	8.4	8.8	0.8
HYBRID-L	15.3	0%	7.6	6.6	1.1
HYBRID-S	15.7	+2.5%	7.8	6.9	1
LINT-D	15.8	+3.2%	7.8	7.1	1
LINT-V	16.1	+5%	7.8	7.3	1
BAYI-11	15.4	+0.6%	7.6	6.6	1.2
BAYI-32	15.4	+0.6%	7.6	6.6	1.1
Politics					
DOM-dLX	18.1	+18.3%	8.6	8.4	1.1
DOM-cLX	16.5	+7.8%	8.2	7.3	1
HYBRID-EU	16.2	+5.9%	8	7.3	0.9
HYBRID-L	15.4	+0.7%	7.6	6.7	1.1
HYBRID-S	15.8	+3.3%	7.8	7	1
LINT-D	16.4	+7.2%	8.3	7.1	1
LIND-V	15.9	+3.9%	7.9	6.8	1.1
BAYI-11	15.4	+0.7%	7.6	6.5	1.2
BAYI-32	15.4	+0.7%	7.7	6.6	1.2

Table 4.6: *Performance of language models on generic domain dataset*

The performance loss is significant for the domain specific models DOM-dLX, DOM-cLX, and HYBRID-EU. The domain-only models that also use a domain-only lexicon lose 20% and 18.3% for biomedical sciences and politics respectively, the HYBRID-EU model 5.9%. As

HYBRID-EU is trained from a significantly larger corpus size, it inevitably contains more vocabulary which explains the significant difference.

As all these models have seen none of the generic data, it is not too surprising that some loss is inevitable. The DOM-dLX implementations already do better than their domain-lexicon only counterparts with relative WER increases of 15.5% (biomedical) and 7.8% (politics).

For models that have shown the most potential in domain-adaptation, the Bayesian interpolation models and the larger hybrid implementations, the loss of generality is more than reasonable ranging between 0% and 0.7%. This means that performance loss is negligible: an ASR engine with one of these models implemented should be safe to use not only when transcribing a narrow domain, but also when exposed to more generic data.

The LINT-D and LINT-V show between 3.2% and 7.2% WER loss, being affected more than other combined models. Considering the mediocre WER performance on either domain, we can suspect that the linear interpolation as executed has simply not found a right probability interpolation between generic and domain n-grams, showing suboptimal performance for each of its parent domains.

Chapter 5

Discussion

In this chapter, we summarize the results from the previous section and analyse their contribution to answering the question how language models can be most effectively implemented in ASR for domain adaptation.

5.1 The impact of language models

The following section provides a discussion of the impact language models in domain adaptation in general as well as the benefits and pitfalls of the different strategies used in this research project.

5.1.1 Best models

The highest obtained word error rate gain for the biomedical domain was by language model **BAYI-32**, showing a 16.3% WER reduction. This gain is composed of 1.1 difference in substitutions (26% reduction), 0.2 difference in deletions (5.4%) and 0.3 difference in insertions (37.5%). For politics, **BAYI-11** is the best with a 4.3% WER reduction, composed of 10.2% substitution-, 1.2% deletion- and no insertion improvement. These results show that domain adaptation in automated speech recognition can indeed be performed by language models.

For both domains, the difference in word error rate between Bayesian interpolated models and hybrid models is small. Considering the fact that the interpolation weights are relatively balanced, this is perhaps not surprising. Both interpolated models have a similar impact on the final n-gram weights, which is comparable to retraining on the combined corpus.

In practice, interpolating language models has the advantage over hybrid training. Having two separately trained domain language models that can be re-used in different domain- or weight configuration can be considered convenient compared complete retraining of models on combined corpora if we imagine a practical implementation of domain adaptation. Given both the high-score in terms of WER and practicality, interpolating a generic language model with a domain-specific language model is therefore the most promising domain-adaptation strategy found in this research. In this, it is important to use an interpolation method such as Bayesian interpolation that does not assume word history independence, since the purpose of this interpolation is in fact to combine two very different domain vocabularies. Choosing Bayesian interpolation is in line with the findings of Pusateri et al. (2019), who proved superiority of Bayesian interpolation above linear interpolation.

The Bayesian- and linear models in this research have been interpolated between the domain language model and the small generic language model GEN-S. While this strategy has

proven successful, interpolating between a larger generic model (BASE) and the domain model can potentially be even better. While such size difference would be of influence in hybrid training, the larger representation of the generic domain n-grams should be only beneficial, as the weighting ratio will continue to give equal weight to the domain-specific language model component. Using the largest available model size should be given priority to using small subsamples when repeating this research. This also applies to the rest of our language models: while we have proven the concept of language model adaptation and the used domain adaptations strategies, it would be interesting to verify these results with larger-sized models.

5.1.2 Reflecting on improvements

Improvements in WER are mainly achieved by reducing the amount of substitutions the model makes. This is expected, as one of the premises of domain adaptation via language model is increasing probability of uncommon domain-specific words to prevent erroneous substitution of such terms with more generic words.

The language model however also appears to have an impact on the deletion rate: for models that do not include any generic data (DOM-dLX, DOM-cLX, HYBRID-EU) deletion rate increases significantly. This effect is present across both domains, although it has relatively more impact in the biomedical domain. The most likely explanation for this given the error samples is that the language modelling data does not contain enough n-grams that reflect the probability of stop-words such as *I*, *the* and *a* in spoken language. Generally speaking, written language is more formal and uttered from a more *objective* viewpoint than speech. Using a domain-only language model as the ones in this project can then cause the probability of jargon being preceded by articles, interjections or pronouns to be underestimated, ultimately contributing to their deletion.

While the deletion rate is definitely affected by language modelling (as is clear given the difference between DOM-dLX and BAYI- models), a significant proportion of the higher deletion rate is likely also caused by failure of the acoustic model. This is especially true for the politics domain, where the deletion rate was already high to begin with. These test files contain a lot of cross-talk and noise compared to the biomedical test set and deletions are present regardless of language model configuration. Another different research design is required to rule out the acoustic model effect (see chapter 6.2).

Language modelling as spelling adaptation

An interesting side effect of language model adaptation is spelling correction. While the generic language model used in the baseline produced American English spelling, integrating the in-domain language model results in a spelling change as can be seen in example 5.1. As the exact number of words that are counted as substitutions but were caused by spelling differences has not been checked, this may have had a distracting influence on WER in our research setup. Even so, using language model adaptation as a form of spelling adaptation might be an interesting practical implementation.

Generic engine transcription: "There has not been enough spent on **defense** for some years." *Pol- transcription*: "There has not been enough spent on **defence** for some years."

Figure 5.1: *Example sentences illustrating spelling adaptation with Poli-BAYI-11*

5.1.3 Data quality

Implementing domain-only language models (DOM-dLX, DOM-cLX) with either an extend or domain specific lexicon has shown a relative increase in word error rate for both domains, deeming it a suboptimal strategy for language modelling in ASR domain adaptation. Especially for the biomedical domain, the subdomain of language model training data has proven too narrow to provide accurate coverage of the test domain, as mainly visible in the rise in deletions. Like discussed in the previous section, this is probably caused by a loss of generality and coverage of spontaneous speech, which the language model training data does not provide.

While the influence of imperfect data coverage was expected for the biomedical domain, the problem also persists for the politics domain. As this corpus is formed of transcripts that are recorded from the same type of meetings as the test data, this was not hypothesized in advance. Despite the fact that the politics corpus is a close match to the test data, the transcripts used for language model training are cleaned. This means that they inherently also represent a more formal category of language. Instead of containing spontaneous utterances, they are reshaped to a pre-written speech-type format. This may influencing the coverage of political debates, which are more spontaneous by nature. Another possible explanation might be simply that the politics domain is selected to be relatively close to the generic domain to begin with, as was confirmed by the perplexity scores of the baseline model. If politics covers a potentially broad range of topics, using a smaller set of data for (domain-specific) language modelling might have a disadvantage over a large generic model, that covers a wider array of topics and speech types.

In terms of corpus size, we have observed a definite advantage of BASE (640 million words of generic data) over GEN-S (4.8 million words of generic data), while HYBRID-S and HYBRID-L (4.8 million vs. 9.6 million) showed relatively similar performance for both domains. Although the size difference between BASE and GEN-S is a lot larger than between HYBRID-S and HYBRID-L, the fact that double the amount of data makes less then 1% difference in relative WER difference for both domains is surprising. We can infer that a larger corpus size of generic data has definite added value in domain-specific ASR (as the probability of covering the domain increases), while size has less impact once a good domain model has already been obtained.

HYBRID-EU, containing a large amount of politics data which was not an exact subdomain match combined with the British politics data, was outperformed by BASE on the politics domain. HYBRID-EU may suffer from the same setbacks as discussed in above: absence of generic, spontaneous speech and formal speech-like transcripts as well as the general disadvantage of the topic-range of politics. While a balanced weighting of British politics data and generic data provided the language model with the power to transcribe British Parliament terms correctly ánd cover generic topics and words, the 55 million words of European Parliament data in HYBRID-EU overpower the British jargon, in addition to still dealing with the formal speech issue.

5.2 Word error rate vs. transcript quality

As discussed, domain-only language models that do not contain generic data show a rise in deletion- and word error rate. Although the introduction of generic data in the BAYI- and HYBRID models seems to solve this problem adequately, there are some points of interest that are worth discussing.

5.2.1 Redeeming the domain-only models

Quantitatively, the rise in WER makes these models *worse* than our baseline model that uses a generic language model. However, as we have seen in figure 4.5 and can again observe in figure 5.2, the readability of a transcript made by such a model is perhaps not worse than the one produced by the baseline engine. Compared to the generic transcript that gets a lot of jargon terms incorrectly, the domain specific model misses out on frequent stop-words more often, but correctly transcribes key terms. When looking from a more qualitative viewpoint, the question is whether WER truly reflects which model is better.

True transcript:
"If you've had an influenza virus, that doesn't stop you getting another type of virus, herpes simplex virus or a human papilloma virus."

Generic engine transcription:
"If you've had an influenza virus, that doesn't stop you getting another type of virus, **a hip** simplex virus or **[DEL]** human **papaloi** virus."

Bio DOM-dLX transcription:
"If you **[DEL]** had an influenza virus, that doesn't stop **your** getting another type of virus, herpex simplex virus or **[DEL]** human papilloma virus."

Figure 5.2: *Example sentences illustrating transcript quality*

Reflecting on our concept of *transcript quality* (3.2.3), it might be more effective in the end to accept the filler word deletions and at least get the content words right. Examining 5.2, both BASE and Bio-DOM-dLX get three errors: BASE does two substitutions on key concepts and one filler word deletion, while DOM-dLX makes one stopword substitution and two filler word deletions. Imagining a situation where one would be provided with either transcript as a subtitle for a biomedical lecture, DOM-dLX would contribute a lot more to understanding what has just been said. A similar effect is also true when we imagine such a situation for politics: when providing live subtitles for a political debate, it would be very questionable to never transcribe party names correctly, even though all other language comes through perfectly.

5.2.2 Alternatives for WER

Observations such as the above call for investigation of alternative error measures: while WER is simple and effective, its equal treatment of content- and stop words is not always an accurate reflection of transcript quality. Ideally, automated speech recognition is not only evaluated and improved on number of errors made, but on the a *weighted* error measure that takes into account some informational value. Nanjo and Kawahara (2005) have proposed such an error measure, *weighted key-word error rate (WKER)* that is based on the *tf-idf* criterion used in information retrieval. Tf-idf, which stands for *term frequency-inverse document frequency* captures the informational value of a word by weighting its importance in a document. If a words occurs frequently in a document, but almost never in other documents in the knowledge base, it is a key term that defines the informational value; if a term occurs frequently over all documents or only once in a document, it is less likely to be a key term and is weighted less accordingly. Weighting each deletion, substitution and insertion by its tf-idf weight delivers more insight in the importance of the error and its effects on understandability than using conventional WER. Other contributions include simple weighting based on *part-of-speech* tag (Garofolo et al., 2000), learning with human-rated weighting scorers to penalize some errors more than others

(Mishra et al., 2011), or the ACE measure that takes into account word predictability and semantic distance to the reference word (Kafle and Huenerfauth, 2017). Improving on the WER statistic is still a relevant topic in ASR resarch (see Future work section).

5.3 Formulating an optimal language modelling strategy

While we have found that language model adaptation is an effective strategy to adapt ASR engines to a topic domain, an interesting question remains if the impact of language model adaptation for a certain topic domain can be predicted. Although language model adaptation is a relatively low-cost strategy, it is clear that some domains benefit more than others. If attempting to transcribe biomedical sciences audio and coming from a generic engine similar to the one used in this project, language model adaptation can be considered a priority, while for politics improvements appear quantitatively less impactful, even though error analysis shows importance for transcribing key concepts.

Ideally, an indication of feasibility of domain adaptation for a certain topic would be achieved by obtaining a measure of *domain distance*. As a proxy of domain distance, we can use the number of new words that is incorporated into the model by expanding the generic lexicon with the vocabulary of the domain-specific corpora. Adding the vocabulary of the biomedical domain to the generic lexicon, we expand with 13,968 new words. For the politics domain, the number of new words is limited to 1,384. This is still a small addition to the original lexicon, which consists of 2271656 words making it a 0.6% and 0.06% growth respectively. This difference of a factor 100 however does not appear to correlate with the difference of factor 4 in word error rate reduction between 16.3% and 4.3%.

Another measure that may give an indication of domain distance is the perplexity of the generic language model on the evaluation data: see 4.1. While the perplexity of the generic model on the biomedical test data set (528.1) is significantly higher than perplexity on the politics dataset (276.1), this measure appears flawed: perplexity on our generic test set is not the lowest, but between politics and biomedical with 375.1. Apart from indicating a larger impact in biomedical domain, nor the lexical distance nor the generic perplexity is explicitly correlated with the chances of reducing the word error rate. Finding such a correlation in order to quantitatively predict impact of language model adaptation is an interesting topic for further research. In order to draw conclusions on this front, a larger study that explores more domains ought to be set up (again see 6.2).

As discussed in section 4.4, *over-adaptation* to a domain only plays a key role in domain-only models that do not incorporate generic data. As soon as generic data is part of the language model, performance loss does not exceed 7.2%, while staying minimal under 0.7% for the best models. Implementation of a model such as BAYI-11 or BAYI-32 is therefore expected to not lose enough generality to make it only applicable to a narrow domain. The *extension* instead of replacing the generic model provides the best of both worlds: being able to capture domain-specific jargon as well as overall generic transcription.

Chapter 6

Conclusion and future work

6.1 Summary

In this research, we have explored the power of n-gram language modelling in domain-adaptation of automated speech recognition engines. Three different approaches to language model adaptation have been tested: training topic-domain specific language models, extending existing generic language models to hybrid models by retraining from a combined domain- and generic corpus, and interpolating between domain-specific and generic language models. A fourth factor taken into account is model size and trade-off between domain match and model size. Word error rate evaluation of ASR engines with different language model configurations was performed for two topic domains: biomedical sciences and politics.

Using a strategy of Bayesian interpolating between domain- and generic fourgram language models, a word error rate reduction of 16.3% for the biomedical- and 4.3% for the politics domain was achieved compared to the baseline model. This shows that topic-domain adaptation of ASR can indeed be performed by language model adaptation. The effect of replacing the baseline generic model with a domain-adapted language model is mostly visible in a significant reduction in the amount of domain-jargon substitutions in the transcripts. By extending the lexicon vocabulary and enhancing the n-gram probability of domain-jargon through an in-domain language model corpus, transcription quality increases significantly.

Language modelling from only in-domain corpora is tricky, due to the fact that it is very important that spontaneous speech n-grams get enough coverage to prevent an increase of deletions. As it is likely that access to spoken, exact topic-match transcripts for specific domains is limited, incorporating generic data with good in-domain data will often be the best approach. Priority should be given to finding proper data that is a close match and provides coverage of the test domain, instead of gathering suboptimal data for a large language model.

Although in-domain language models still improve the understandability of transcripts by getting domain jargon right, they show an increased deletion rate that can be easily prevented by extending the generic lexicon constraint instead of replacing it and using a proportion of generic data in training. This can either be the hybrid training approach from a combined corpus or Bayesian interpolation, but the latter is preferred. Linear interpolation is not the best choice for combining two domains with very different vocabularies, given its assumption of word history independence across domains.

6.2 Future work

In this thesis, we only cover the impact of language model adaptation compared to a generic baseline model under the assumption that language modelling is an effective way to perform domain adaptation. Although this assumption is more than affirmed, we can not exclude the possibility that adapting the acoustic model would have influence to some extent as well: the difference in effect between adapting language models and acoustic adaptation remains unknown. In order to draw conclusions about the degree to which either the language models or acoustic model contributes to e.g. deletion rate, this research should be extended with experiments that compare language model only domain adaptation to an approach where the acoustic model *is* retrained with domain-specific data. A comparison between acoustically adapted models, language model only adapted models and a fully adapted ASR pipeline on the task of topical domain adaptation should give more insight to which components contribute to which errors most heavily, solidifying the grounds to choose either language model adaptation or a different approach for domain adaptation.

Due to limits in time and data availability, only two domains have been covered in language model adaptation research. Although the successes and optimal approach found can be regarded as a proof of concept, verifying this domain adaptation strategy in more domains and languages would be an interesting topic for further research. More specifically, a quantitative prediction of language model domain adaptation impact could be of great interest for many practical ASR users. Since language model adaptation is a relatively low-impact strategy in terms of data scarcity and computational resources, this could potentially allow for easier and accessible improvement of individual speech recognition engines. In order to estimate the impact of of a refined language model, we would ideally find some type of correlation between a measure of domain distance and transcript quality- or word error rate improvements. As this research has only considered two domains, no such correlation could be retrieved. A quantitative analysis of the benefits of our method of domain adaptation is therefore also a line or research worth exploring.

This research has only considered n-gram language models, which are still the most widely implemented type of language models in automated speech recognition. As mentioned in chapter 2 however, *neural language models* have shown great potential for improving word error rate in ASR (Raju et al., 2019) and can be applied in domain-adaptation (Raju et al., 2019; Liu et al., 2021). The added computational complexity in the pipeline is the main reason practical implementations still use n-grams. As more and more research also takes interests in ways to circumvent this issue and take benefit of the neural language model revolution, extending our domain adaptation task to neural language models is of course also interesting. As neural language models take into account unlimited context history to begin with and already improve significantly on n-gram language models, a comparative study similar to this thesis would be interesting to answer the question in how far domain adaptation would still be feasible, or if one large language model would provide sufficient coverage overall.

Finally, the insights obtained in the error analysis of our results have reconfirmed the limits of how informative word error rate is in evaluating quality of ASR transcripts. If or when repeating this project setup, an interesting angle would be to use another measure for transcript quality that takes into account understandability.

Chapter 7

Appendix A: evaluation data

Reference	Duration	Speaker(s)	Attribution
Biomedical science education			
Cell anatomy	04:44	F en-GB	Doctor Me Clever (2018a)
Subcellular structures	04:34	F en-GB	Doctor Me Clever (2018b)
Immunity introduction	14:31	M en-GB	John Campbell (2011a)
Immunity types	10:20	M en-GB	John Campbell (2011b)
Neurobiology	51:26	F en-ZA	MIT OpenCourseWare (2014)
Stem cells	46:03	M en-US	MIT OpenCourseWare (2020)
Cancer	1:04:22	M/F en-US	Harvard Medical School (2019)
DNA	28:05	M en-US	Khan Academy (2009)
Telomeres	18:46	F en-AU	TED (2017)
Politics			
House of Commons 1	18:42	M/F en-GB	British House of Commons (2022)
House of Commons 2	20:42	M/F en-GB	British House of Commons (2022)
House of Commons 3	14:20	M/F en-GB	British House of Commons (2022)
House of Commons 4	15:15	M/F en-GB	British House of Commons (2022)
House of Lords 1	13:01	M/F en-GB	British House of Lords (2022)
House of Lords 2	11:20	M/F en-GB	British House of Lords (2022)
House of Lords 3	10:30	M/F en-GB	British House of Lords (2022)
House of Lords 4	10:53	M/F en-GB	British House of Lords (2022)
Children Committee 1	19:11	M/F en-GB	Children and Families Act 2014 Committee (2022)
Children Committee 2	15:43	M/F en-GB	Children and Families Act 2014 Committee (2022)
Children Committee 3	23:28	M/F en-GB	Children and Families Act 2014 Committee (2022)
Treasury Committee 1	15:10	M/F en-GB	Treasury Committee (2022)
Treasury Committee 2	15:19	M/F en-GB	Treasury Committee (2022)
Treasury Committee 3	22:19	M/F en-GB	Treasury Committee (2022)
Treasury Committee 4	15:30	M/F en-GB	Treasury Committee (2022)

Table 7.1: *(Extrinsic) evaluation data for the biomedical science education and politics domains*

Bibliography

R. K. Aggarwal and M. Dave. Acoustic modeling problem for automatic speech recognition system: conventional methods (part i). *International Journal of Speech Technology*, 14(4): 297–308, 2011.

C. Anoop, A. Prathosh, and A. Ramakrishnan. Unsupervised domain adaptation schemes for building asr in low-resource languages. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 342–349. IEEE, 2021.

A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33: 12449–12460, 2020.

T. C. Bell, J. G. Cleary, and I. H. Witten. *Text compression*. Prentice-Hall, Inc., 1990.

J. R. Bellegarda. An overview of statistical language model adaptation. In *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.

Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

British House of Commons. House of commons wednesday 30 march 2022, 2022. URL https://parliamentlive.tv/event/index/0589be74-81b5-40cf-910d-48d602510046#player-tabs. [Online; accessed 20-April-2022].

British House of Lords. House of lords thursday 7 april 2022, 2022. URL https://parliamentlive.tv/event/index/8376d81c-f23f-4b6c-9677-40a0aefade12?in=11:07:46#player-tabs. [Online; accessed 20-April-2022].

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394, 1999.

Children and Families Act 2014 Committee. Children and families act 2014 committee monday 4 april 2022, 2022. URL https://parliamentlive.tv/event/index/d3b8e048-8f59-485e-bf18-c7a93b450df9#player-tabs. [Online; accessed 20-April-2022].

Doctor Me Clever. Cells, aqa 9-1 gcse biology, topic 1 cell biology, 2018a. URL https://www.youtube.com/watch?v=f5-3sCyBzJY&t=6s. [Online; accessed 20-April-2022].

Doctor Me Clever. Subcellular structures, aqa 9-1 gcse biology, topic 1 cell biology, 2018b. URL https://www.youtube.com/watch?v=9AIVFjPmrUY&t=8s. [Online; accessed 20-April-2022].

J. S. Garofolo, C. G. Auzanne, E. M. Voorhees, et al. The trec spoken document retrieval track: A success story. *NIST SPECIAL PUBLICATION SP*, 500(246):107–130, 2000.

W. Ghai and N. Singh. Literature review on automatic speech recognition. *International Journal of Computer Applications*, 41(8), 2012.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264, 1953.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

Harvard Medical School. Cancer metabolism: From molecules to medicine, 2019. URL https://www.youtube.com/watch?v=2wseM6wWd74. [Online; accessed 20-April-2022].

K. Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W11-2123.

B.-J. Hsu. *Language modeling for limited-data domains*. PhD thesis, Massachusetts Institute of Technology, 2009.

B.-J. P. Hsu and J. Glass. Style & topic language model adaptation using hmm-lda. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 373–381, 2006.

D. Hwang, A. Misra, Z. Huo, N. Siddhartha, S. Garg, D. Qiu, K. C. Sim, T. Strohman, F. Beaufays, and Y. He. Large-scale asr domain adaptation using self-and semi-supervised learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6627–6631. IEEE, 2022.

H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe. Transfer learning of language-independent end-to-end asr with language model fusion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6096–6100. IEEE, 2019.

H. Jeffreys. *The theory of probability*. OUP Oxford, 1998.

F. Jelinek. Interpolated estimation of markov source parameters from sparse data. In *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

John Campbell. Immunity, 2011a. URL https://www.youtube.com/watch?v=ZNXvHqnl-_U. [Online; accessed 20-April-2022].

John Campbell. Immunity 1, introduction, specific and non-specific immunity, take 1, 2011b. URL https://www.youtube.com/watch?v=bo_TNSF3-dw. [Online; accessed 20-April-2022].

W. E. Johnson. Probability: The deductive and inductive problems. *Mind*, 41(164):409–423, 1932.

R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

S. Kafle and M. Huenerfauth. Evaluating the usability of automatically generated captions for people who are deaf or hard of hearing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 165–174, 2017.

A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar. An analysis of incorporating an external language model into a sequence-to-sequence model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5828. IEEE, 2018.

S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3): 400–401, 1987.

Khan Academy. Dna, 2009. URL `https://www.youtube.com/watch?v=_-vZ_g7K6P0`. [Online; accessed 20-April-2022].

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing*, volume 1, pages 181–184. IEEE, 1995.

P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86, 2005.

A. Lee, T. Kawahara, and K. Shikano. Julius—an open source real-time large vocabulary recognition engine. volume 3, pages 1691–1694, 01 2001.

G. J. Lidstone. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8(182-192):13, 1920.

L. Liu, Y. Gu, A. Gourav, A. Gandhe, S. Kalmane, D. Filimonov, A. Rastrow, and I. Bulyko. Domain-aware neural language models for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7373–7377. IEEE, 2021.

V. Manohar, P. Ghahremani, D. Povey, and S. Khudanpur. A teacher-student learning approach for unsupervised domain adaptation of sequence-trained asr models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 250–257. IEEE, 2018.

T. Mishra, A. Ljolje, and M. Gilbert. Predicting human perceived accuracy of asr systems. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

MIT OpenCourseWare. 24. neurobiology 1, 2014. URL `https://www.youtube.com/watch?v=dKLkXQEN9XU`. [Online; accessed 20-April-2022].

MIT OpenCourseWare. 24. stem cells, apoptosis, tissue homeostasis, 2020. URL `https://www.youtube.com/watch?v=EJ6Sjn1c04Y`. [Online; accessed 20-April-2022].

H. Nanjo and T. Kawahara. Unsupervised language model adaptation for lecture speech recognition. In *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

H. Nanjo and T. Kawahara. A new asr evaluation measure and minimum bayes-risk decoding for open-domain speech understanding. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I–1053. IEEE, 2005.

H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38, 1994.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

E. Pusateri, C. Van Gysel, R. Botros, S. Badaskar, M. Hannemann, Y. Oualil, and I. Oparin. Connecting and comparing language model interpolation techniques. *arXiv preprint arXiv:1908.09738*, 2019.

A. Raju, D. Filimonov, G. Tiwari, G. Lan, and A. Rastrow. Scalable multi corpora neural language models for asr. *arXiv preprint arXiv:1907.01677*, 2019.

B. Roark, R. Sproat, C. Allauzen, M. Riley, J. Sorensen, and T. Tai. The OpenGrm open-source finite-state grammar software libraries. In *Proceedings of the ACL 2012 System Demonstrations*, pages 61–66, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/P12-3011.

TED. The science of cells that never get old — elizabeth blackburn, 2017. URL https://www.youtube.com/watch?v=2wseM6wWd74. [Online; accessed 20-April-2022].

Treasury Committee. Treasury committee wednesday 30 march 2022, 2022. URL https://parliamentlive.tv/event/index/df312789-eea9-40d4-9cda-f03f46abd58d#player-tabs. [Online; accessed 20-April-2022].

W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx-4: A flexible open source framework for speech recognition, 2004.

D. Wang, X. Wang, and S. Lv. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018, 2019.

I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *Ieee transactions on information theory*, 37(4):1085–1094, 1991.

S. Wotherspoon, W. Hartmann, M. Snover, and O. Kimball. Improved data selection for domain adaptation in asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7018–7022. IEEE, 2021.

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al. The htk book. *Cambridge university engineering department*, 3(175):12, 2002.