Karlsruhe Institute of Technology

Enhancing Multilingual Language Models with Language Meta-Information

Bachelor's Thesis
of

Michael Wiegner

At the Department of Informatics
Institute for Anthropomatics and Robotics (IAR)
Interactive Systems Lab (ISL)

Primary referee:	Prof. Dr. Alex Waibel
Secondary referee:	Prof. Dr. Tamim Asfour
Advisor:	Dr. Jan Niehues

Duration: June 14th 2017 – September 13th, 2017

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text, and have followed the rules of the KIT for upholding good scientific practice.

Karlsruhe, September 13$^{\text{th}}$, 2017

Michael Wiegner

Abstract:

Machine translation has been a heavily studied topic over the last half-century. In recent years, research in the field has been once again reinvigorated thanks to progress in neural networks. Among recent developments is a push towards multilingual neural network language and translation models, to increase performance while cutting down on the amount of models needed to maintain. In this work we examine the benefits of supplementing a multilingual language model with high-level linguistic information. Using the database from the World Atlas of Language Structures (WALS), we create a vector for each language, containing the presence or absence of certain linguistic features, and use them to set the initial states of our model. Compared to a multilingual model without these custom initial states, perplexities ended up being over 10% lower, while overall performance remained close to the monolingual baseline. In a character based setting, when learning a new language with limited resources, the multilingual models managed to achieve a lower perplexity than monolingual models. Furthermore, after training in a new language, the model using the WALS vectors forgot less of the previously learned languages than a model having had its initial states set by only using numerical IDs.

Kurzzusammenfassung:

Maschinelle Übersetzung ist ein Forschungsgebiet, das im letzten halbem Jahrhundert ausführlich untersucht wurde. In den letzten Jahren wurde die Forschung in dem Gebiet signifikant wiederbelebt, dank Fortschritten im Bereich von neuronalen Netzen. Unter anderem sieht man eine steigende Zuneigung zu multilingualen neuronalen Sprach- und Übersetzungsmodellen, um die Leistung zu verbessern und insgesamt weniger Modelle trainieren zu müssen. In dieser Arbeit untersuchen wir welche Vorteile das Ergänzen eines multilingualen Sprachmodells mit linguistischen Informationen für die Qualität hat. Wir benutzen den World Atlas of Language Structures (WALS) um für jede Sprache einen Vektor zu erstellen, der die An- und Abwesenheit bestimmter linguistischen Eigenschaften kodiert, und benutzen diesen um die Anfangszustände unseres Modells zu setzen. Im Vergleich mit einem multilingualem Sprachmodell, das nicht diese angepassten Anfangszustände benutzt, haben wir um 10% niedrigere Perplexitäten erreicht, während die allgemeine Leistung nahe der monolingualen Ausgangswerte war. Beim Lernen von einer neuer Sprache mit wenigen Ressourcen, schafften es Multilinguale Modelle auch bessere Perplexitäten zu erreichen als monolinguale Modelle, in einem charakter-basierten Ansatz. Das Modell, das die mit WALS erstellten Sprachvektoren benutzt hat, hat weniger von den alten Sprachen die es davor erlernt hatte vergessen als ein vergleichbares Modell, dessen Sprachvektoren mit einfachen Sprachidentifikationsnummern initialisiert wurden.

Acknowledgements:

I would like to thank my advisor Dr. Jan Niehues for his help and reassuring guidance, which proved invaluable while completing my thesis.

My thanks also goes to my family and friends, for their encouraging words and continued support.

Contents

1. Introduction

1.1. Introduction

It is not hard to imagine why the field of machine translation has seen a considerable amount of research and human effort invested into it over the last several decades. Designing a machine that would be capable of effectively and correctly translating input from one human language into another is a challenging and noble goal of great value to society. The existence of such a machine would be an enormous step towards, among other things, the elimination of language barriers that exist between different peoples, and the problems that come with them.

The idea of computers being able to achieve this linguistic prowess is not too inconceivable. As anyone who has translated a text could attest, re-expressing a sentence in another language often seems like a very rote task, which is exactly the kind of work computers are traditionally good at. As is often the case however, the devil is in the details.

After several cycles of breakthroughs and setbacks in the field of statistical machine translation, research has been once again reinvigorated in recent years, mainly due to the popularization of neural networks, their effectiveness in modelling the task at hand, as well as the democratization of the computational ability required to train them.

While neural networks themselves come in a number of shapes and sizes, one thing they all have in common is the high amount of data needed to train them. For machine translation, often one (neural network based) model is used for translating from one specific language to one other specific language, requiring a considerable amount of data from all language pairs to be able to translate from and to any language.

An alternative approach to these one-to-one specialized models was the development of multilingual models, where a single model would get inputs in more than one pair of languages. Through exposure to multiple languages, the model would be able to learn more about languages and translation in general, develop an internal interlingua, and improve translations across the board using transfer knowledge, with less data needed per specific language pair compared to a one-to-one model. Recent findings seem to support this theory [18] [20] [19].

In any case, training these networks is usually done through vast amounts of example sentences, with little in terms of higher level information about the languages themselves. This is possibly wasted potential, as there are undoubtedly higher-level relationships and linguistic commonalities between different languages, and knowledge of them could be a means of improving language modelling and translation.

In this work we shall examine a possibility of supplementing multilingual language and translation models with information about the different languages they are handling, through the use of higher-level structural data about languages.

- **Chapter 2** serves as an introduction to the various topics breached by this work. Among these are the fundamentals of statistical machine translation, neural networks and how they are jointly used to create the language modelling systems at the heart of our experiments.

- **Chapter 3** gives an overview of relevant work that has been done on neural language modelling and translation, especially character based language modelling and multilingualism in translation models, inspiring the line of research pursued in this work.

- **Chapter 4** goes into the motivation behind the experiments and elaborates on our approach of incorporating high-level linguistic information into language and translation models.
- **Chapter 5** describes the specifics of the experiments, elucidates the reasoning behind the decisions taken and presents the experimental results.
- **Chapter 6** puts the results into context and proposes avenues of future research.

2. Background

In this work, extensive use is made of a number of ideas, concepts and metrics that are very specific to statistical machine translation on the one hand, and neural networks on the other. In order to fully understand some of the hypotheses advanced here, it is essential that there be at least a basic understanding of relevant elements in both these disciplines, especially language modelling, recurrent neural networks or perplexity.

The purpose of this chapter is to act as a concise presentation of this prerequisite knowledge.

2.1. Statistical Machine Translation

Before going into the subject matter itself, we shall first take a look at a more formal definition of what we want to achieve.

The task of machine translation can be described as the transformation of a sentence in the input (source) language $S = s_1, \ldots, s_{|S|}$ into a sentence in the output (target) language $T = t_1, \ldots, t_{|T|}$. A machine translation system could therefore be modelled by a function

$$\hat{T} = translate(S) \tag{2.1}$$

where $\hat{T}$ is a translation hypothesis of source sentence S by the system.

More specifically, in Statistical Machine Translation (SMT), one looks at the problem in terms of probabilities[1]. As such, we see $P(T|S)$ as the probability of translating sentence S into T. With Bayes' theorem, this can be expanded to

$$P(T|S) = \frac{P(S|T) \cdot P(T)}{P(S)} \tag{2.2}$$

so that $P(T|S)$ is now expressed in relation to the independent probabilities of sentences S and T, as well as the probability that T is a translation of S. In order to give the best possible value to our translation hypothesis $\hat{E}$ in this system, we should find the sentence T which is the «most likely» translation [1] of sentence S:

$$\hat{T} = arg \max_T P(T|S) = arg \max_T \frac{P(S|T) \cdot P(T)}{P(S)} \tag{2.3}$$

This can be further simplified to

$$\hat{T} = arg \max_T P(T|S) = arg \max_T P(S|T) \cdot P(T) \tag{2.4}$$

since the input sentence S is given. The original problem is now split into two smaller ones: $P(T)$ is determined by the language model, whereas $P(S|T)$ is determined by the translation model.

The success of a SMT system is therefore very dependent on how well the probabilities above are able to be estimated. These are usually determined through analyses of very large amounts of sample data, most notably parallel text corpora in the source and target languages, aligned on a sentence level.

2.2. n-gram Language Modelling

The task of language modelling is in many ways a distinct one from translation, as we only need to take one language into account. It is through it that the probability $P(T)$ of a sentence T is determined. One of the main uses of a language model is to use it to evaluate how natural or fluent a given sentence sounds in that language. With such a model, it would be possible to evaluate and thus improve the output quality of a translation system, by taking into account how natural the proposed sentence sounds.

In more formal terms, when we want to calculate $P(T)$ for a sentence $T = t_1, \dots, t_L$ we can define it as

$$P(T) = P(|T| = L, t_1, \dots, t_L) \tag{2.5}$$

which is the probability of the sentence length being L, the first word being t_1, the second being t_2 etc. and the last word being t_L.

Often this very intricate joint probability of the sentence is rewritten as a product of word probabilities. An additional «end of sentence» (<eos>) token is added to the end of the hypothetical sentence in order to be able to incorporate the probability of its length. In mathematical notation:

$$P(T) = \prod_{i=1}^{L+1} P(t_i|t_1, \dots, t_{i-1}) \tag{2.6}$$

where the $L+1^{th}$ word is the <eos> token. To illustrate this concept, let us look at the example sentence «I like trains». The probability of this sentence is $P(|T| = 3, t_1 =«\text{I}», t_2 =«\text{like}», t_3 =«\text{trains}»)$, which, if transformed with the equation above is equal to $P(t_1 =«\text{I}») \cdot P(t_2 =«\text{like}»|t_1 =«\text{I}») \cdot P(t_3 =«\text{trains}»|t_1 =«\text{I}», t_2 =«\text{like}») \cdot P(t_4 =\text{<eos>}|t_1 =«\text{I}», t_2 =«\text{like}», t_3 =«\text{trains}»)$.

This long chain of multiplication of probabilities is however still quite unwieldy. A relatively simple and straightforward way to estimate these probabilities is used in the so-called n-gram model. The basic idea relies on applying the Markov assumption, to the modelling of sentences.[10] Only the $n-1$ last words are taken into account when computing the probability of the next word. Therefore

$$P(T) = P(t_1, \dots, t_{L+1}) \approx P(t_1) \cdot \ldots \cdot P(t_{L+1}|t_{\max(1,L-n-1)}, \dots, t_L) \tag{2.7}$$

2.3. Neural Networks

Over the years, significant progress has been done in developing various structures of neural networks, each with its special characteristics and properties. These serve as basic building blocks for many modern applications of machine learning. Some notable mentions are feed forward neural networks, recurrent neural networks, time delay neural networks [21] and convolutional neural networks. In this section, we shall look at the neural network types most often used in the scope of neural machine translation.

2.3.1. Feed Forward Neural Networks

Artificial neural networks are a structured mass of connected artificial neurons, conceptually similar to those we have in our brain. These networks consist of an input and output layer, along with an arbitrary number of hidden layers. Neurons in adjacent layers are connected through directed weighted edges to neurons in the next closest layer to the output layer. Feed forward neural networks are universal function approximators, meaning that they are able to approximate any function with any precision, provided that the size of the hidden layer is large enough.[15]

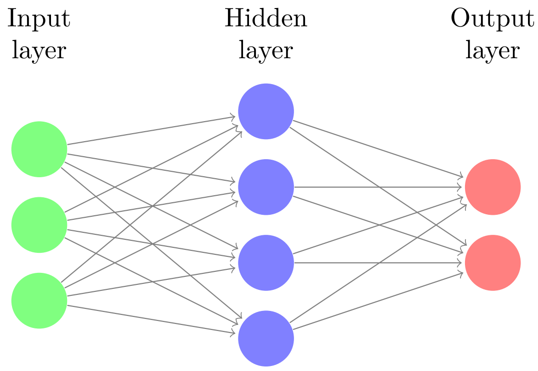


Figure 2.1.: Visual example of a feed forward neural network. Based on [7]

More formally, the output $o_{(i,j)}$ of neuron i of layer j is $f_j(\sum_{k=1}^{n} w_{(k,i,j)} \cdot o_{(k,j-1)})$, where f_j is the activation function for layer j (like the sigmoid or tanh functions for example), $w_{(k,i,j)}$ is the weight of the connection from neuron k on layer $j-1$ to neuron i on layer j. The whole network can be seen as a complex nested function, with the input flowing through the various layers up to the output layer. Optionally, a trainable per-neuron constant factor called bias can be added to the input of the activation function.

These neural networks are trained using an optimization method in a process called backpropagation [13]. To be able to train a network, one needs sample data of inputs and expected outputs. The inputs are fed into the network, and the outputs are then compared to the expected outputs. By using a cost function, such as cross entropy or mean squared error, we can determine the error E, and then determine by going through the network backwards how much a weight w contributed to the error. To reduce the error, one can calculate the gradient of E in regards to w, and then modify the weight for example in a process called gradient descent:

$$w_{t+1} = w_t - \eta \cdot \frac{\partial E}{\partial w_t} \tag{2.8}$$

where η is the learn rate parameter.

To train a neural network into a usable state, often large quantities of data and training steps are needed.

2.3.2. Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network which contains one or more feedback loops. This means that the previous output of the neural network influences what the network will output on the next input, hence recurrent. Unlike feed forward neural networks, recurrent neural networks can give different outputs for the same input over time. This makes this type of neural network especially good at modelling sequences. Since natural language is essentially a sequence of sounds, sentences, words or characters, it isn't hard to understand how they have found themselves to be used for language modelling and machine translation.[16]

Figure 2.2 illustrates on how the inputs and outputs are linked through time in a basic RNN cell.

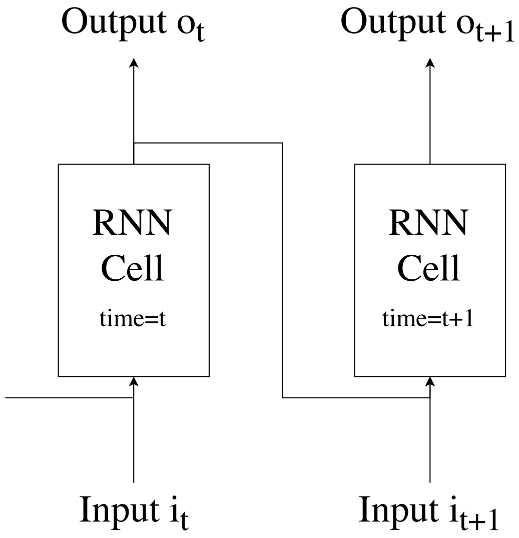


Figure 2.2.: Interaction of Outputs and Inputs in Recurrent Neural Network cells.

At timestep $t+1$, the output o_{t+1} is not only dependent on input i_{t+1}, but also on the previous output that has been generated in timestep t, o_t. Both o_t and i_{t+1} are concatenated and then input into the RNN cell, giving the output o_{t+1}. This output would then be used along i_{t+2} to get o_{t+2} and so on.

2.3.3. Long Short Term Memory

The vanishing gradient problem[14] is an issue that arises during backpropagation with gradient descent. As a neural network gets deeper, it becomes harder to train, because after going back multiple layers in backpropagation, the contribution of a specific node to the error rate gets closer and closer to zero, hindering weight updates. To train recurrent neural networks, they need to be unrolled through time first, since a previous output of the network influence the next one. In many ways this becomes similar to training a very deep neural network.

Long Short Term Memory (LSTM)[5] is a special recurrent neural network architecture which is more resilient in regard to the vanishing gradient problem. This means that it is well suited to learn dependencies and correlations over a larger timespan, something which classic recurrent neural networks have trouble with. In fact, remembering things for the longer term is often set as the default behaviour of a LSTM cell[6]. This is done by an internal cell state. With every step, information from the previous cell state can be forgotten, and depending on the previous output and current input, new information can be added to it. These modifications to the cell state are done by linear operations, which simplifies the flow of long term information and allows backpropagation through time to take them better into account.

2.4. Evaluating Language Models

When using a dataset to train a language model, one of the most important things from the get go is to split it into three parts: training data, validation (development) data and test data. The training data is used to train the model parameters, the validation data is used to adjust model hyper-parameters and the test data is used for the evaluation of the final model.

Once we have our test data, we still need to have an objective measure to determine the quality of a language model, how accurately it can actually manage to model the desired language. One such measure is perplexity.[10]

Our test data set $\mathcal{T}$ is composed of sentences T and our language model can give us $P(T)$. We can therefore look at the probability of the whole test data set $P(\mathcal{T})$ as the product of the probabilities of the individual sentences of the test set $\prod_{T\in\mathcal{T}} P(T)$, since the sentences are considered independent. The perplexity of the test data for a given language model is then defined as follows:

$$perplexity(\mathcal{T}) = exp(-\frac{logP(\mathcal{T})}{length(\mathcal{T})}) \tag{2.9}$$

where $length(\mathcal{T})$ is the amount of elements (words) in $\mathcal{T}$, i.e. $\sum_{T\in\mathcal{T}} |T|$. In information theory, $-\frac{logP(\mathcal{T})}{length(\mathcal{T})}$ is called entropy.

For a more intuitive understanding of this measure, one can imagine perplexity as the average number of times one would have to draw a word from the probability distribution to get the correct one. A lower perplexity means that the model is more «sure of itself» concerning which word is to come next, differentiating it from a purely random distribution. In short, for the purposes of training language / translation models, a lower perplexity is better.

2.5. Sequence to Sequence Models

A sequence to sequence model is one of the more common arrangements of RNN cells that are used. As its name implies, the construction is made to, given a sequence, generate another sequence. Since translation is also just generating a target sequence (sentence) from an source sequence, neural machine translation systems often employ a variation of this to do their translation.

A basic sequence to sequence model is made out of two parts: the encoder and the decoder. The encoder receives the input sequence step by step, generating no output. What is being changed during this process is the hidden state of the encoder. Once the source sequence has been fully input into the encoder, the encoder state is then transplanted into the decoder. The decoder then receives a special input token to give its first output of the sequence. In subsequent steps, it is customary that the decoder is fed the last word it has output, until it itself outputs the end-of-sequence token.

A visualization of this process can be seen in figure 2.3

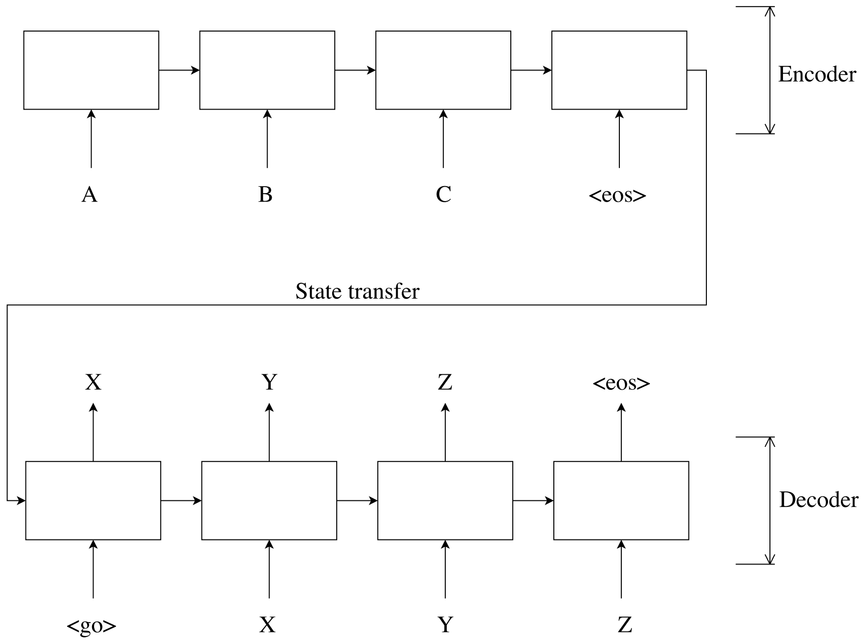


Figure 2.3.: A basic sequence to sequence model. The encoder and decoder steps are unrolled through time.

2.6. Beam Search

When a sequence to sequence model is being used for translation, the decoder gives the probability of the next element (word, character, etc.) in the output sentence one at a time. However, the best theoretical translation is the most probable output sequence as a whole. Doing a complete search and calculating the probability of all plausible sentences is a far too resource intensive and impracticable approach. Conversely, always greedily choosing the next most probable element in the sequence is bound to deliver sub-optimal results. A compromise between the two lies with using beam search.

With beam search, the n most probable hypotheses are considered, n being defined as the beam width [10]. The beams themselves can be thought of as a path along a series of branches in an ever-growing probability tree. At every step of adding a new element to the active output sequence hypotheses, the n most probable sequences overall are determined and kept, extending the some of the beams, and ignoring those that didn't make the cut. Using this technique often leads to better translations, at a moderate (and customizable) performance cost.

2.7. Byte Pair Encoding

The issue of having to make do with a limited vocabulary is quite obvious. With less words available to it, the language or translation model is limited in the amount of words it can learn and reproduce, which then places a limit on the quality of translation that can be achieved. This can be even more limiting for languages that have grammatical cases and declension, since every time a noun is seen in a different case, it would be counted as a separate word.

One approach to counter this is the usage of byte pair encoding[12][11]. With byte pair encoding, analysis of the input text is done to determine which sub-strings in words appear most frequently, and to keep those together, while splitting up other parts of words and marking these parts with a special marker, signifying that it is part of an unfinished word. Apart from decreasing the vocabulary size, the goal of this approach is for example to split complicated words into smaller, potentially independent and reusable semantic units, or separate the radix of a word from the various suffixes encountered. An example of such a separation can be seen in figure 2.4.

$$\text{air@@}\left\{\begin{array}{l}\text{man}\\\text{lines}\\\text{less}\\\text{time}\\\text{ports}\\\text{ship}\\\text{waves}\\\text{ing}\\\text{bags}\end{array}\right.$$

Figure 2.4.: Examples of word endings for word component «air»

The letters «air» are followed by @@, a character sequence that doesn't appear in the original text. It is used to distinguish the word component «air» (which is supposed to be followed by the remainder of the same word) from the independent word «air». For all the words in the example, the split position of the words done by the byte pair encoding seems quite natural, as «air» is a semantically meaningful piece of a word, and one could easily imagine that some of the suffixes are common enough to be reused in words such as fireman or friendship.

With byte pair encoding, how far the splitting up of words into sub-words will go is very configurable, as it depends directly on the desired size of the vocabulary as well as the amount of unique words in the original text.

3. Related Work

A considerable amount of progress has been done over the last few decades in neural networks for them to have become a powerful enough tool to use for effective neural machine translation. Consequently, research in the field of neural machine translation is more lively than ever, and often sees breakthroughs every few months.

Several attempts have been made to look into whether there are benefits to using multilingual language and translation models. The obvious advantage of such an approach would be that one multilingual model could replace multiple ones destined to specific language pairs. To be a viable alternative however, this multilingual model would need to at least show performance parity to language pair specialized models.

A number of recently published research articles have shown promising results in that regard. Sharing the attention mechanism between translation models while training with multiple language pairs has lead to performance improvements, especially for low-resource language pairs[19]. Similar observations about multilingualism were made in other work, where it also improved performance for low-resource languages, and showed potential for translation between languages which were not present as a direct language pair at the time of training. [20].

A paper from Google researchers [18] proclaimed that they had fully embraced multilingual models after seeing the benefits multilingualism had for them. They too found that multilingualism offered a way to increase performance for languages where little data is available. Furthermore, they managed to have the model do zero-shot translation, meaning that the source and target language pair had not been trained explicitly by the model. There seemed even to be evidence of an implicit interlingua being created. These results made them decide to henceforth employ multilingual models in their production environments.

The last few years have also seen improvements and inventive methods in to alleviate the problem of having a limited fixed vocabulary, by looking at sub-word units and/or characters. For example, methods have been developed to efficiently cut vocabularies down to size, deciding on where to split which words based on algorithms such as byte pair encoding [12].

Then there are the various character-based approaches to neural machine translation and language modelling, with research often focusing on different ways to approach the character/word barrier in order to see performance improvements. This means for example using character based inputs and outputs, internally transforming character sequences into words [24] [25] [26]. Another example was to use a subword-level encoder and character-level decoder [23]. A slightly different approach was a hybrid model using a character based backup solution when confronted with a rare word [22].

Improving performance through incorporation of linguistic data has also been researched for some topics in statistical machine translation. One paper saw an improvement in the ability to do dependency parsing in low-resource languages by using the a subset of the information in The World Atlas of Language Structures (WALS) related to the existence of part of speech classes in certain languages and their relative positions [28]. In another paper, multilingual word alignment was used to automatically do typological studies of language structure, and showed large similarities to related WALS data [30]. In word reordering, WALS has been also employed to split languages into different categories [31]. Information from WALS was also made use of in creating a one-to-many reordering model using a feed forward neural network [29].

Making use of language similarities in machine translation has been also researched as a follow up to word embeddings. In one approach, similarities between languages were exploited to enhance dictionaries and phrase tables used in statistical machine translation [27]. First, word embeddings were built from two sets of large monolingual data, one for each language.

Then, the limited amount of bilingual data is used to estimate the linear projection that exists between the embedding spaces. Using this knowledge, correct translations could be estimated through extrapolation.

4. Enhancing Multilingual Language Models with Language Meta-Information

4.1. Motivation

Multilingualism in language and translation models can be both a blessing and a curse. On one hand, being exposed to multiple languages can help it attain more general knowledge about how languages work, and extrapolate things learned in one language to another. On the other hand, a multilingual model has to prove that it can compete with monolingual models, which are specialized in their language, in terms of quality for it to be viable. This is something easier said than done, as being fed data from multiple languages can also lead the model to being overwhelmed and less sure about which word in which language it is supposed to output. It is therefore essential that the model be set up in a way that it will be able to make use of its multilingualism in a productive manner. It should be able to find the common ground many languages share, but also be aware of their specificities and differences.

In this work, we aim to help a multilingual model achieve to find these commonalities and differences by using the extensive database of the World Atlas of Language Structures, a compilation of a large number of linguistic meta-information about languages. To examine this, we shall examine the performance of language models trained using this additional information. We shall then observe how easily a multilingual model with this information is able to learn a new language, after having previously been trained in a large number of languages along with information from the WALS.

4.2. Extracting Information from the World Atlas of Language Structures

The World Atlas of Language Structures [2] is an extensive database of various kinds of higher-level structural meta-information about languages. First published in 2005 as a book with a CD-ROM, it has since been published online and gradually expanded and improved, and is now under the supervision of the Max Planck institute for Evolutionary Anthropology. The language features documented within it cover a wide range of areas, such as phonology, morphology, information about nominal categories, nominal syntax, verbal categories, verbal syntax, word order in sentences, information about how simple and/or complex clauses are formed etc. These features have been curated into the Atlas from a variety of descriptive materials, such as reference grammars, and each data point from each feature has a direct reference to that source.

An example to help illustrate the nature of the contents of the Atlas would be feature 33A: Coding of Nominal Plurality, in the “Nominal Categories” area. Put in simpler terms, the coding of nominal plurality is the way in which one would express the plural of a noun in that language. For example in English, it is expressed via a plural suffix. The plural of computer is computers, with “-s” being the plural suffix. In other languages things are done differently however, with prefixes, stem changes, or even complete reduplication, such as in Indonesian. This means that the same word is used for both singular and plural, so “computer” or “computers” would still just be “komputer” in Indonesian.

Another example, this time from the “Word Order” area would be feature 87A: Order of Adjective and Noun. This documents whether nouns are followed or preceded by their adjectives, or if there is no dominant order. While English has the “Adjective-Noun” order, French for example has the “Noun-Adjective” one. One of the most obvious signs of this difference can be seen with common abbreviations between English and French. While one would say D.N.A. in English, which stands for deoxyribonucleic acid, in French it is A.D.N., which stands for acide désoxyribonucléique. The simple word order switch here is easily apparent.

A total of 192 features are documented in the World Atlas of Language Structures, for a total of 2679 languages. One thing to note with these numbers is that not every language has every feature documented, something related to the amount of available descriptive materials for any specific language.

In addition to linguistic data, the entries in the Atlas also hold geospatial information about the languages, so that different data points for a feature can be visualized on a map. An example of this can be seen in Figure 4.1.

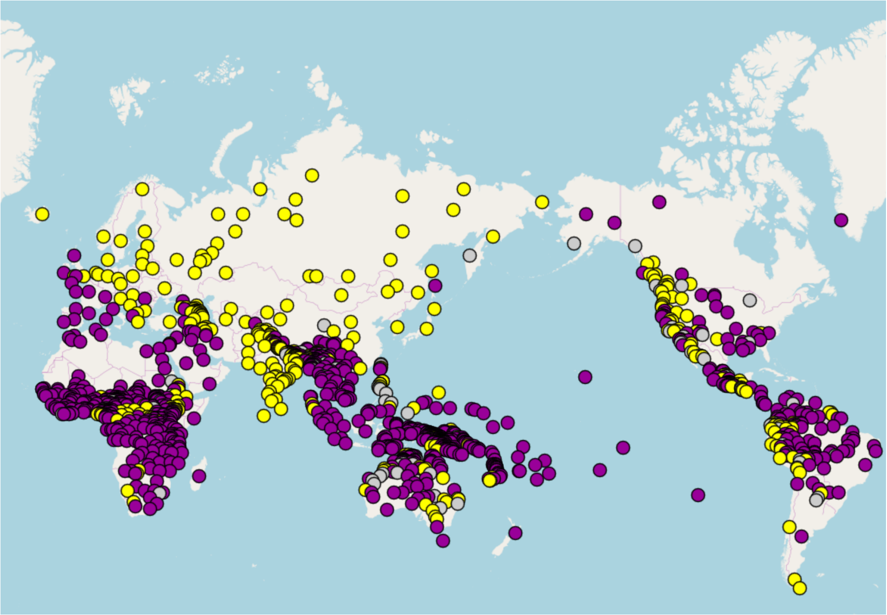


Figure 4.1.: Feature 87A: Order of Adjective and Noun visualized on a world map. The yellow dots represent languages with the «Adjective-Noun» order, the purple ones with the «Noun-Adjective» order and the grey ones represent the languages where there is «No dominant order», as well as all other remaining data.

The entire database of the WALS can be viewed on their website [2] or downloaded as a CSV file. It contains every entry available for the 192 features in 2679 languages the database holds, along with geographic and language genus and family data. Most of the feature data are of categorical nature, so the information to be gained when encoding the various values of features is whether or not a certain feature is present or not.

In practice this leads to all the features of a language being encoded into a single language vector as shown in figure 4.2.

Since the value for feature 32A is the second value out of nine, the second bit out of a nine bit portion in the language vector is set to one. With this encoding method, the total length of the language vector is 1138 bits.

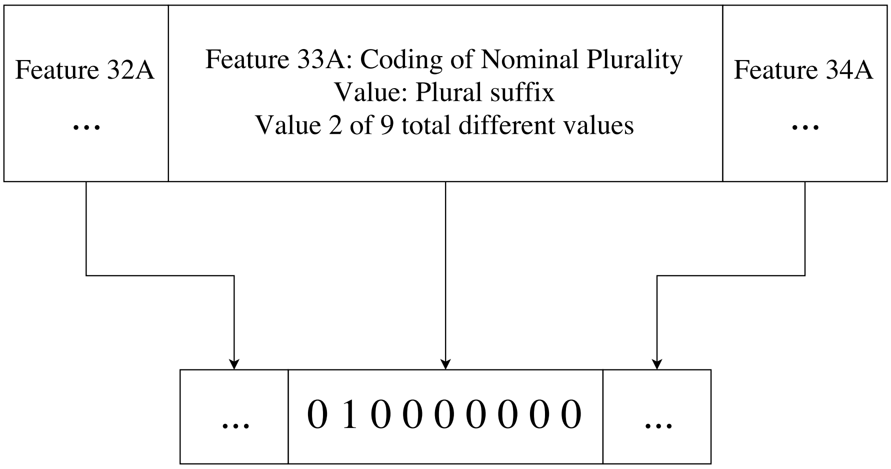

Feature 32A
...
Feature 33A: Coding of Nominal Plurality
Value: Plural suffix
Value 2 of 9 total different values
Feature 34A
...
...
0 1 0 0 0 0 0 0 0
...

Figure 4.2.: Example of WALS feature encoding for French

4.3. Incorporating WALS Information into Language Models

When thinking about feeding additional information into a neural network, it is essential to think about where in the network to feed this information into. With the information being which language with which features the model will be faced with next, it makes sense to incorporate that information as soon as possible into the network. It is also important that the information be incorporated in a way in which the model will be able to learn to use it, and interpret it for itself. A place that meets both these requirements in the network is the initial state of the LSTM cells in the network.

Figure 4.3 shows how the language vectors would be incorporated into a sequence to sequence model. By using the language vector to set the initial state, the model can make use of the language information from the get-go, from the first word of the sentence onwards. Of course, in this scenario, how the language vector itself is turned into an initial state for the neural network should also be trainable. Since language and translation models have multiple layers, not every information from the language vector would be useful everywhere, and it should be up to the network which information should be stored and used in which location.

More specifically, the language vector is fed into a feed forward neural network (with bias), with an output layer the size of all the necessary states to initialize. The output vector is then split and the various states are initialized with the obtained values. Two LSTM layers are represented in 4.4, as that was the depth of the neural network used. This setup allows the network to train what information about the input language from the language vector it stores in which layer and part of the initial network state. The C state in a LSTM is the one used for long-term memory. The H state usually contains the previous output, but since we are conveying the input language information before any output has been made by the network, we can use it as well for our initialization purposes.

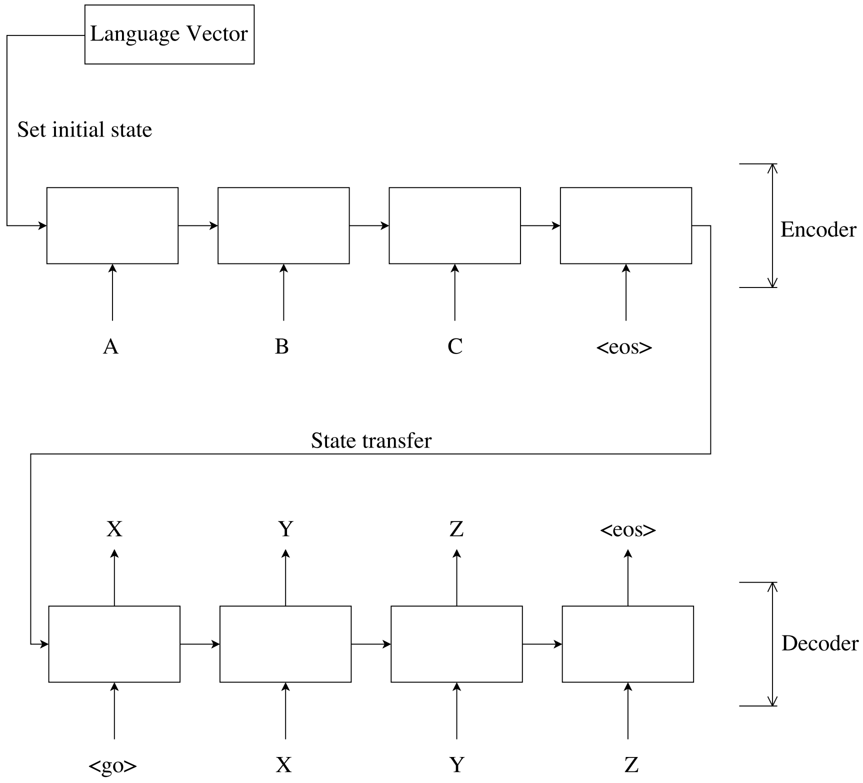


Figure 4.3.: Enhancing a standard sequence to sequence model with language vectors. The encoder and decoder steps are unrolled through time.

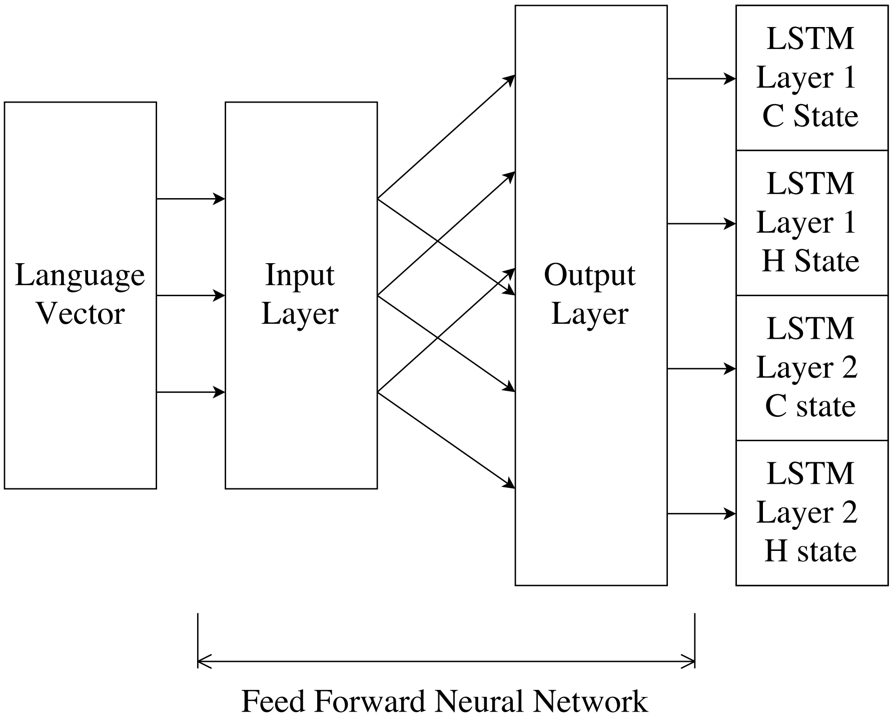


Figure 4.4.: Transforming the language vectors into the initial states of the LSTM layers.

5. Evaluation

5.1. Experimental Setup

The language data used for training, validating and testing the models came from the corpus of transcribed and translated TED Talks, held in the Web Inventory of Transcribed and Translated Talks. [3][4]. This data was then tokenized and cleaned. During cleaning operations, the text was transformed into lower-case characters, numbers were replaced by the place-holder capital letter N, information for the hearing impaired, such as notifications about applause or musical cues, were removed, as well as the constant repetition of speaker names during dialogues. The dataset was split into 120k sentences for training, and 10k sentences each for validation and testing. These numbers were chosen so that for every language chosen we can have the same amount of sentences.

The basis for the sequence to sequence model used was tf-seq2seq[8], essentially a re-implementation of the standard sequence to sequence model used for machine translation in the official TensorFlow tutorials[9], incorporating features from the new TensorFlow API. This model was then ported to Python 3, and its input and training pipelines were modified to suit the needs of the experiment.

The reason for choosing a sequence to sequence model to modify for this task is because of its inherent ability to be used as both a language model and a translation model, depending on whether the input sequences into the encoder are empty or not.

Two types of language vectors were used in experiments. The first kind were vectors containing all the information the WALS database had on the languages. The second kind were one-hot encodings of the language ID. This allows us to see the impact the WALS data itself has on the language model, separately from the fact that information about which language is being input into the model at the beginning.

Apart from this modification, the neural network saw no particular changes made to them. Training was stopped when the perplexity over the validation data saw no improvement for several epochs. Care was taken to train models in the same category with the same general training parameters, to allow for accurate comparisons.

5.2. Word-Based Language Modelling with 4 Languages

The first experiment was done with 4 languages: Czech, German, English and French. A total of 7 models were trained, four of them monolingual, and three multilingual ones. All of them used the same vocabulary, generated through byte pair encoding, using subword-nmt[1], with a size of 40,000. The vocabulary was not coded on a language specific basis, i.e. the byte-pair-encoded word chunks were shared across languages.

Among the three multilingual language models, one was fed the language vectors based on the information from the World Atlas of Language Structures, one received vectors with a one-hot encoding of the language ID, and the third one was trained without any language vectors, lacking an input pipeline for them altogether.

The results can be seen in table 5.1.

[1] `https://github.com/rsennrich/subword-nmt`

	Monolingual	Multilingual WALS	Multilingual IDs	Multilingual No Vectors
Czech	140	149	151	170
German	102	105	106	118
English	77	80	81	91
French	65	67	68	75

Table 5.1.: Individual perplexities of the three multilingual language models for each of the 4 languages, with perplexities of monolingual models as a reference

Of the multilingual language models, the one with the best perplexities is the one that used the WALS language vectors, but only by a slight margin compared to the one using the language IDs. Both are however noticeably better than the model which used no language vectors, their perplexity being 11-13% smaller for every one of the four languages. On the other hand, the models were not as good as the separate monolingual models, albeit not by much. The perplexities of the vector enabled models were only 2-6% larger than the ones of the monolingual models.

Overall, this seems to indicate that the addition of language vectors into the language model, and thereby telling the model in which language the next sentence will be through the initial state, helps a multilingual language model perform better, similarly to what can be seen with separate monolingual ones. However, there doesn't seem to be any strong implication as to whether the information distilled from the World Atlas of Language Structures gave the model an advantage over the one using language IDs. Even during training, the validation perplexities of the two models were neck and neck to each other.

One of the reasons why the difference between the language vectors could be so small is simply that the WALS vectors could be underused, that the multilingual model simply isn't multilingual enough. With four languages, it could be sufficient to learn to differentiate between them and learn what to put into the initial state of each language independently. There would then be no significant advantage to having information indicating commonalities between languages in that case.

To look into this line of reasoning, we decided to test this hypothesis in the next experiment.

5.3. Character-Based Language Modelling with 13 Languages

In order to potentially see more of an impact from WALS information, this experiment was done with 13 different languages: Czech, German, English, Spanish, French, Croatian, Dutch, Polish, Portuguese, Romanian, Russian, Turkish and Ukrainian. These languages were partly chosen due to them having various similarities but also dissimilarities between them. For example French, Spanish, Portuguese and Romanian are Romance languages. Then there are a number of Slavic languages, like Czech, Croatian, Polish, Russian and Ukrainian, of which the last two even have a different character set from all other languages in this list. Then there are also languages which are the sole representatives of their genus here, such as Turkish. But the language vectors contain more than just the «superficial» information about the genus, as linguistic commonalities between languages run deeper.

One fundamental change compared to the previous model is that it is now character based. This means that the language model is no longer modelling a language and its sentences on the level of words, but rather on the level of characters. This makes it easier to realize these multilingual models, as creating a vocabulary with both a reasonable size but also enough words in the languages wouldn't have been an easy task. It also allows us to reuse the model to learn an additional language, but more on that in the next section. The total amount of different characters, including punctuation and white space ended up being slightly above 300. This is also why perplexities are not comparable with those in the previous section, as here they are computed on a per-character basis.

A total of 15 Models were trained. 13 of them were monolingual ones, one for each language, forming the performance baseline in terms of perplexity for the two multilingual models. For each language 120,000 sentences of training data were used, resulting in a training data set of 1,560,000 sentences for the multilingual models. As in the previous experiment, they make use of language vectors, one of the WALS based ones, the other the language IDs. Unlike the previous experiment however, a model without any vectors was not trained, as it seemed relatively clear that it would not be able to perform as well as the other two multilingual models.

The results can be seen in table 5.2.

	Monolingual	Multilingual WALS	Multilingual Language IDs
Czech	3.50	4.15	4.15
German	2.92	3.39	3.39
English	2.98	3.37	3.38
Spanish	2.92	3.28	3.28
French	2.66	3.09	3.10
Croatian	3.35	3.95	3.94
Dutch	3.06	3.53	3.53
Polish	3.32	3.91	3.90
Portuguese	2.97	3.33	3.33
Romanian	3.18	3.68	3.68
Russian	3.23	3.77	3.76
Turkish	3.22	3.73	3.72
Ukrainian	3.36	3.93	3.92

Table 5.2.: Individual perplexities of the multilingual language models for each of the 4 languages, with perplexities of monolingual models as a reference

At first glance, it is obvious that the multilingual models have perplexities 14-20% higher than the results of the monolingual ones. What is also nearly identical perplexities between the two multilingual models. The differences are always no bigger than one hundredth of a perplexity point, and go both ways, so there is no clear superior. All in all, both models perform just as well, for better or worse. There once again is no clear distinction between the two performance-wise.

Since the neural network has a long time over several epochs to learn the data, it is probable that even if the model exposed to the WALS vectors did learn more about what binds and separates the languages it is learning, the ID based model would simply figure out how to model these languages just as well on its own. In fact, for the first half-epoch of training, the model with the WALS vectors had a better training and validation perplexity. Once this first half-epoch were passed, the models were indistinguishable from a performance standpoint.

This then begs the question, in what contexts can the information gained from WALS be effectively used? What the WALS vectors certainly do offer is a level of connection between languages that isn't there for the language IDs. If both models were presented with a previously unseen language, the ID based model would have to learn an appropriate initial state for it from scratch, whereas the model with the WALS vectors would have already several points of reference, depending on what linguistic features the new language had or hadn't. Consequently, the abilities of the previously trained network of learning a new language is the focus of the next experimental series.

5.4. Zero-Shot Language modelling

When deciding on languages to use for the experiment, the idea was to pick a language that already shares a number of similarities to the previous languages the model has seen, as well as one that is relatively different from it. That way, the manner in which similarity and dissimilarity is picked up on through the language vector information can be investigated. Italian and Hungarian were the perfect candidates. Italian is, like 4 of the languages the model has previously learned, a Romance language. There should therefore be enough previous experience with that language type to discern what does and does not belong to it. Hungarian on the other hand is known as a relatively unique language, as it is very different from other European languages. It is the only major language of the Ugric genus.

A number of experiments were made with different vocabulary sizes and compositions in order to evaluate how well the models would fare with potentially having less data in their language to learn.

Before looking at any of the results of the models which have received supplemental training, let us first take a look at the performance of the models as they are, in table 5.3.

In the subsequent tables, WALS and IDs refer to the fully trained 13 language multilingual models, which have been trained with WALS language vectors and ID language vectors respectively.

5.4.1. Analysing Existing Performance in Untrained Languages

	WALS (untrained)	Language IDs (untrained)
Italian	12.37	12.71
Hungarian	41.56	42.38

Table 5.3.: Perplexity of multilingual models on evaluation sets in never before seen languages

The two multilingual models which showed so little difference in performance for the original 13 languages now show that they are distinct, albeit only slightly. The WALS model is slightly better that the ID based one, but both have seen so much of Romance languages that the difference remains minimal. Unsurprisingly, the perplexity for Hungarian is far higher than the one for Italian in both cases. The WALS model is again better than the language ID based one here, but again not by a significant amount.

To get a closer and more detailed look at the current state of the networks, we can let them generate some of their most probable sample outputs. This will allow us to take a glimpse what kind of sentence structures have been impregnated into the network. It will also allow us to discover what the networks think of the language vectors of the two languages they haven't seen yet, as one can ask the network to generate input from the resulting initial states. To get several different samples, a beam search of beam width 10 is carried out. The results can be seen in the tables in appendix A. Two are additionally reproduced here for illustration purposes.

By analysing these tables, it is hard to miss that the most common recurring phrase generated in all the trained languages is «thank you». This is closely followed by other sentence snippets one would expect to hear a lot of in a TED talk, such as «this is» or «why». A noteworthy observation is that the language vectors definitely seem to set up the model to give outputs in that language, as the language of the sample sentence always matches the language of defined by the language vector.

The really interesting part of these tables however is what they show about the handling of the two new languages, Italian and Hungarian, by the language model. The first few outputs for Italian and Hungarian are relatively short, and not very characteristic of any of the trained languages in particular. In fact, the first few outputs of Hungarian and Italian are the shortest outputs the network has produced. After that, the sentences become somewhat longer, but usually remain nonsensical, and not really clearly tied to one specific language. For example in 5.5, the sample text of the language ID model for Hungarian is «new york times don't simple personne», which is an odd mixture of English of French, and the name of the «new york times» itself could come from any of the languages. The language ID based Italian as well as the WALS

Language	Sample Text (WALS)	Language	Sample Text (IDs)
Czech	děkuji.	Czech	děkuji.
German	und das ist eine andere.	German	vielen dank.
English	thank you.	English	thank you.
Spanish	así que esto es lo que hacemos.	Spanish	esto es lo que hacen.
French	merci.	French	merci.
Croatian	to je ono što možemo.	Croatian	to je bilo N.
Hungarian	all.	Hungarian	ok.
Italian	ok.	Italian	no.
Dutch	dank u.	Dutch	dit is een voorbeeld.
Polish	dziękuję.	Polish	dziękuję.
Portuguese	o que é que estamos a fazer?	Portuguese	obrigado.
Romanian	de ce?	Romanian	de ce?
Russian	это не так.	Russian	спасибо .
Turkish	teşekkürler.	Turkish	teşekkürler.
Ukrainian	дякую.	Ukrainian	дякую.

Table 5.4.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 1.

based Hungarian fared not much better. The exception here is the WALS based Italian. After some short sentences, it starts elaborating a long sentence in Portuguese.

Taking into account the language proximity, this might not seem too unexpected. After all Italian and Portuguese are both Romance languages, but as can be seen in table 5.6, they share only 50 features, about 61% of the total amount of features in Italian. Feature-wise the closest language to Italian is Spanish, with 69 features, covering 85% of Italian features, and the second closest one is French, with 63 features, or 77%. However, from the perspective of the Portuguese language vector, Italian shares 84% of its features, which is the highest similarity to any other language it has. Also, no other language has quite as much overlap proportionally as Portuguese with Italian by a significant margin, the next closest values being 56% for both Romanian and Ukrainian. This could then explain the production of Portuguese text.

The heat map 5.1 gives a visual representation of the feature proximity of the language vectors. It is asymmetric, as not all the languages in the World Atlas of Language Structures contain the same amount of documented features. This means that while Russian shares 53 of the 55 features Ukrainian has, which is 96% of the total, for Russian, 53 features represents only 34% of its 155 features. The way to read the heat map is in a column-like fashion, i.e. the similarity of the language of the row and the language of the column is expressed in relation to the total amount of features of the column language.

Looking at the column for Italian for example gives us the values closest to Italian, which, as mentioned previously, are Spanish and French, along with English and Russian as well. It is by looking at the row for Italian, i.e. how similar the Italian language vector is to other languages, that it becomes clear why Portuguese text was generated with the Italian language vector, as it obviously stands out with its 84%. It also makes most sense that this is also the way the model would react in while attempting to create associations between languages, by comparing how similar the new vector is to the existing ones, instead of how similar other vectors are to it.

In the same line of reasoning, we can also try to look at the row for Hungarian in the heat map, and see why inputting the Hungarian language vector in the WALS model failed to generate text in a consistent language. What is instantly noticeable with the Hungarian vector is that there are no high outliers in terms of similarity as was the case for Italian. The languages with the highest similarity are Ukrainian and Russian, at a meagre 60%, and these don't even use the same characters. On the flip side, there doesn't even seem to be a language that strongly dissimilar to it, the lowest being Dutch at 40%.

Language	Sample Text (WALS)
Czech	takže to je to , co se stane , že jsem se podíval na to , že jsem se na to podíval.
German	aber das ist eine andere geschichte ist.
English	thank you very much of.
Spanish	así que esto es lo que estamos haciendo algunas personas que están en el mundo.
French	c'est ce que nous avons fait.
Croatian	to je ono što možemo napraviti nešto.
Hungarian	muchas generatives.
Italian	então , o que estamos a construir uma coisa que estamos a construir uma coisa que estamos a trabalhar para as pessoas que estamos a trabalhar para os mesmos.
Dutch	het is een andere manier van de mensen.
Polish	powiedziałem : " nie...
Portuguese	portanto , o que é que estamos a fazer?
Romanian	acesta este un exemplu despre cel mai mare.
Russian	это очень просто , который мы начинаем.
Turkish	teşekkürler....
Ukrainian	я не знаю , що це було не так.

Language	Sample Text (IDs)
Czech	takže jsem se stalo na to , co se stalo , že je to velmi dobrý.
German	und das ist eine großartige versuche.
English	thank you very much , you know , that's what we do it.
Spanish	esto es lo que hacer.
French	nous avons besoin de la main.
Croatian	to je bilo jednostavno , ali ne možemo učiniti da je to napraviti.
Hungarian	new york times don't simple personne.
Italian	alternativa , design , designer.
Dutch	dit is een voorbeeld van de verschillende staten.
Polish	dziękuję bardzo , że nie wiedziałam.
Portuguese	porque é que estamos a fazer isso?
Romanian	acesta este un moment de această persoană a fost într-un fel de mare.
Russian	я не знаю , что это не так.
Turkish	teşekkürler!
Ukrainian	але це не так?

Table 5.5.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 9.

Language	# features identical with Italian	# features identical with Hungarian	out of
Czech	24	28	57
German	49	77	156
English	60	87	158
Spanish	69	80	154
French	63	72	157
Croatian	28	30	59
Dutch	39	36	88
Polish	43	48	88
Portuguese	50	29	59
Romanian	46	46	82
Russian	57	94	155
Turkish	30	85	153
Ukrainian	31	33	55
Hungarian	39	154	154
Italian	81	39	81

Table 5.6.: Number of shared features with Italian and Hungarian from the other 13 Languages

One of the languages most dissimilar to all others in this language set is Turkish, yet Hungarian has a similarity factor of 55% to it, and ends up being the language most similar to Turkish. With all the similarity factors for Hungarian being relatively evenly spread between 40 and 60%, it is no wonder why the model had no clear sense what to associate with it. It simply was a language not too similar or dissimilar to other ones, even though at 154 language features the vector isn't exactly sparse.

Overall, it seems that the usage of inter-related language vectors gave the WALS model a very slight advantage over the one with language IDs in terms of anticipating what the new language would be like, especially if it was more strongly related to previously seen languages. However, this ability is clearly somewhat limited, and dependent on how much other similar languages have been seen, as well as the amount of information that was available from the World Atlas of Language Structures for that specific language.

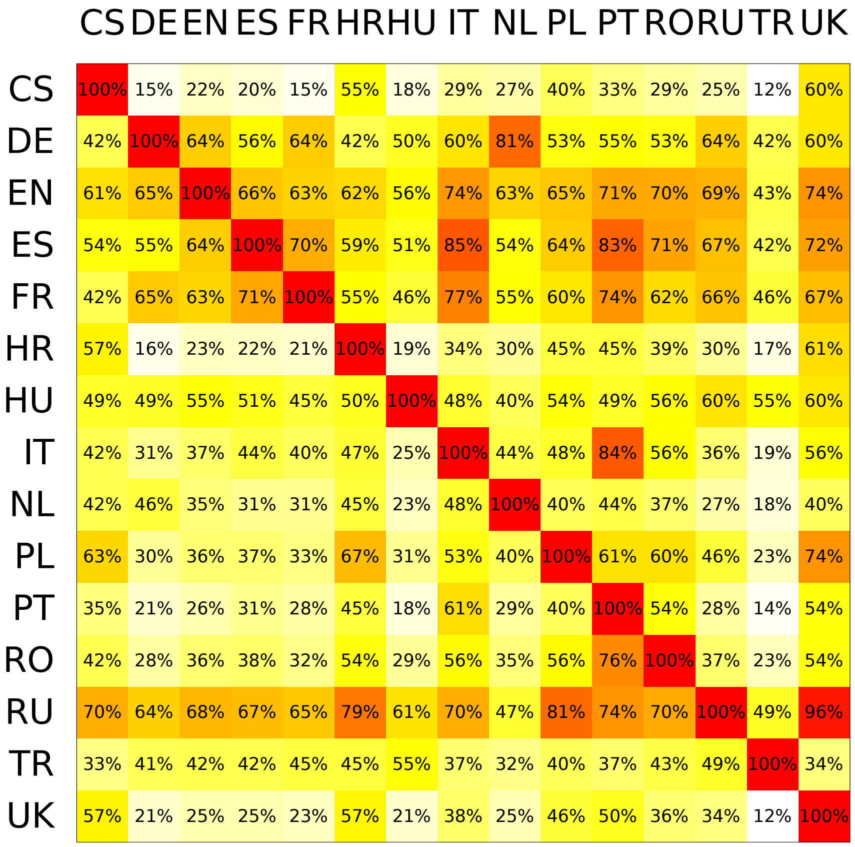
5.4.2. Training With New Language Data

After having analysed how the networks trained in 13 languages perform with previously unseen languages, we shall look at what happens once they do get trained with the new language data of either Italian or Hungarian.

In the first experiment performed, the full size of the training datasets of 120,000 sentences was used. Model parameters were kept the same for consistency purposes, and to be able to compare it to values from table 5.2, where the multilingual character based models were compared to monolingual models. A monolingual baseline for the two new languages was also set up for this experiment. The results can be seen in table 5.7.

	Monolingual	WALS + 120k	IDs + 120k
Italian	2.88	2.95	2.95
Hungarian	3.20	3.37	3.37

Table 5.7.: Perplexities of multilingual models trained with the complete dataset, compared to monolingual models.



	CS	DE	EN	ES	FR	HR	HU	IT	NL	PL	PT	RO	RU	TR	UK
CS	100%	15%	22%	20%	15%	55%	18%	29%	27%	40%	33%	29%	25%	12%	60%
DE	42%	100%	64%	56%	64%	42%	50%	60%	81%	53%	55%	53%	64%	42%	60%
EN	61%	65%	100%	66%	63%	62%	56%	74%	63%	65%	71%	70%	69%	43%	74%
ES	54%	55%	64%	100%	70%	59%	51%	85%	54%	64%	83%	71%	67%	42%	72%
FR	42%	65%	63%	71%	100%	55%	46%	77%	55%	60%	74%	62%	66%	46%	67%
HR	57%	16%	23%	22%	21%	100%	19%	34%	30%	45%	45%	39%	30%	17%	61%
HU	49%	49%	55%	51%	45%	50%	100%	48%	40%	54%	49%	56%	60%	55%	60%
IT	42%	31%	37%	44%	40%	47%	25%	100%	44%	48%	84%	56%	36%	19%	56%
NL	42%	46%	35%	31%	31%	45%	23%	48%	100%	40%	44%	37%	27%	18%	40%
PL	63%	30%	36%	37%	33%	67%	31%	53%	40%	100%	61%	60%	46%	23%	74%
PT	35%	21%	26%	31%	28%	45%	18%	61%	29%	40%	100%	54%	28%	14%	54%
RO	42%	28%	36%	38%	32%	54%	29%	56%	35%	56%	76%	100%	37%	23%	54%
RU	70%	64%	68%	67%	65%	79%	61%	70%	47%	81%	74%	70%	100%	49%	96%
TR	33%	41%	42%	42%	45%	45%	55%	37%	32%	40%	37%	43%	49%	100%	34%
UK	57%	21%	25%	25%	23%	57%	21%	38%	25%	46%	50%	36%	34%	12%	100%

Figure 5.1.: Heat map of how many percent of features language vectors share. Asymmetry is due to the different amounts of documented features for each language.

The perplexities achieved by the multilingual models seem to be much closer to those of their monolingual counterparts than in the large multilingual example. Here, their perplexities are only 2 - 5% larger than those of the monolingual models. Unfortunately, finding a performance difference between the two models remains as difficult as ever. When analysing the training logs, both are again neck and neck in terms of training and validation perplexity.

With such a large training set, both models essentially have the opportunity to learn the new language from scratch if they so desired. After all, the monolingual model did it and has the best perplexity. This then begs the question of how much using a reduced dataset would have a performance impact across the board.

In the next experiment, the size of the training data was severely reduced to only 1% of the original size, at 1200 sentences. Results of training the multilingual models, as well as of a new monolingual baseline, can be seen in table 5.8.

	Mono 1.2k	WALS + 1.2k	IDs + 1.2k
Italian	5.74	4.42	4.41
Hungarian	6.87	5.81	5.76

Table 5.8.: Perplexities of multilingual models trained with a very small dataset.

First of all, perplexities are higher across the board, which is to be expected with so much less training data, the monolingual baseline being almost multiplied by two. This time however, the multilingual models perform better than the monolingual ones, in both languages. Their perplexities are lower by about 17-22%. It seems as though exposure to multiple languages can improve language modelling capabilities in cases where little training data in the desired target language is available. This characteristic has been already noted in other related work. The difference between the multilingual models is however relatively small, with the language ID based one being slightly better.

To put the experiment into perspective, a variation with a slightly larger reduced dataset was performed, this time being 10% of the original input size, at 12,000 sentences. The results can be seen in table 5.9.

	Mono 12k	WALS + 12k	IDs + 12k
Italian	4.24	3.48	3.48
Hungarian	4.87	4.20	4.19

Table 5.9.: Perplexities of multilingual models trained with a mid-sized dataset.

While the perplexities are overall lower than in the experiment with the smaller dataset, they still do not reach those of the models with the large datasets. Here again, the perplexities of the multilingual models are 14-18% lower than the monolingual baseline. Between the multilingual models there is again little difference to be seen.

So far, the models trained have all received only sentences in the new language they were supposed to learn. Since the models exposed to reduced data seemed to profit from having had previous experience with other languages, it wouldn't be inconceivable that by adding some sentences of these previously seen languages into the reduced training set, the model might make better use of information about modelling other languages it has within.

In the subsequent experiment, the 1200 sentence dataset was extended by 100 sentences for each of the 13 languages previously seen by the multilingual models. This means that the proportion of old language sentences to sentences in the new language is approximately one to one. The results of the training can be seen in table 5.4.2.

	WALS + (1.2k + 13 × 100)	IDs + (1.2k + 13 × 100)
Italian	4.60	4.58
Hungarian	6.22	6.16

Table 5.10.: Perplexities of multilingual language models trained on a small dataset, including sentences from the previous 13 languages.

Surprisingly, the perplexities are about 5% higher than when training only using new data. It seems that the addition of additional sentences in other languages proved more distracting than helpful.

To see if this is a trend, enhancement in the same proportions was done on the 12,000 sentence dataset.

	WALS + (12k + 13 × 1000)	IDs + (12k + 13 × 1000)
Italian	3.53	3.54
Hungarian	4.25	4.26

Table 5.11.: Perplexities of multilingual language models trained on a mid-sized dataset, including sentences from the previous 13 languages.

While this time, the difference is only of about 1-2%, the same negative impact of adding more sentences in other languages seems to be at work. While the language ID based model had a slightly better perplexities than the WALS based one in the former of the two experiments, the opposite is true in the latter experiment.

After having trained different multilingual models exclusively in a new language, it might be interesting to take a step back and take a look at how much of the old capabilities of being able to model 13 different languages are still present afterwards. This might offer some more insight into how the multilingual models learn the new language, and how they deal with their previous knowledge while doing it.

The multilingual models trained with 120,000 sentences have been re-evaluated on the test sets of the original 13 languages they were trained on. The resulting perplexities are shown in table 5.12.

	WALS + 120k IT	IDs + 120k IT	WALS + 120k HU	IDs + 120k HU
Czech	13.64	21.93	34.66	35.44
German	9.06	13.06	13.84	14.12
English	5.80	6.92	9.86	8.93
Spanish	10.71	10.75	14.66	15.08
French	10.33	11.93	12.32	12.86
Croatian	11.31	13.50	21.02	20.14
Dutch	9.49	9.99	13.53	13.12
Polish	13.74	22.53	27.76	30.45
Portuguese	12.00	11.56	14.90	15.59
Romanian	11.23	16.38	17.29	19.00
Russian	5.72	7.24	7.45	8.14
Turkish	11.61	20.74	31.50	32.50
Ukrainian	5.91	8.19	8.20	8.94

Table 5.12.: Re-evaluation of model performance in original 13 languages after having learned Italian or Hungarian

The resulting values are surprisingly quite different from model to model. While the performance itself has dropped considerably across the board, there are significant differences in terms of performance loss in the between models. Even though the WALS and Language ID models had almost identical results in terms of how well they performed on the languages they trained for, it seems that the way they learned the new language and preserved their old capabilities was different.

In terms of learning Italian, the ID based model has a worse performance in all but one language than the WALS one, and often this performance drop is considerable. For languages like Czech, Polish or Turkish, the perplexity is nearly double that of the WALS model, for others it is almost 50 % more. The languages with the most similar perplexities between the two seem to be Spanish, French, Portuguese and Dutch. This might be due to the fact that Italian is related to at least the first three of these languages. Overall performance wise, the most unaffected languages seem to be Russian and Ukrainian, their perplexity having gone up only by about 50% for the WALS + 120k IT model compared to what the same model could before having learned Italian. For other languages, the perplexity has usually gone up multiple-fold. This is probably because both Russian and Ukrainian use Cyrillic characters, which no other language makes use of.

Taking a look at the models that have learned Hungarian, one sees a similar pattern. Overall the perplexities of these models are higher than those of the ones trained in Italian. Between the two Hungarian models, the one having been trained with WALS vectors has better perplexities in 10 of the 13 languages, but the differences are not so pronounced as in the previous case. Learning Hungarian, a language dissimilar to the ones previously learned, has taken a far larger performance toll on the model's previous language abilities than another Romance language like Italian.

In general, it appears that the models using WALS are less destructive of their previously acquired knowledge and capabilities than the models that use language IDs. Nevertheless, a large factor in how these models are able to perform after being trained in only one language greatly depends on how similar or dissimilar it is to the languages already learned.

6. Conclusion

6.1. Summary

In this work we looked at a way how we could use meta information about languages in order to enhance language models with them and show them the various kinds of links and similarities that exist between languages.

We first extracted all the linguistic information available in the World Atlas of Language Structures and turned it into vectors for every language, documenting its features. We then modified a language model in order to allow for the language vectors to be fed through a simple feed forward neural network, whose output is used to initialize the model's LSTM cells.

We then looked at the how this model performed in 4 languages compared to one using language vectors consisting only of a unique language identifier, as well as one without any vectors. We were able to conclude that receiving vectors indicating the coming language leads to a better perplexity, comparable to the one of the monolingual baseline. However, no significant difference could be detected between using a model with language features in their language vector or a simple id. In further experiments with 13 languages, no such difference could be detected either, the models always performed more or less the same.

When examining the capabilities of the two models, previously trained in the 13 languages, in their ability to be able to learn a new language, the model using the language feature vectors had a slight advantage. It was able to create an association to other similar languages through common features being present in the language vector. This didn't last for very long into the training cycle however, and both ended with comparable performances.

Multilingual language models with more experience with languages can get significantly better results than a monolingual one in case of a small training set in the new language.

Interestingly, the model using the language features didn't forget its old language capabilities as fast as the one with the language identifiers did during training for a new language.

In conclusion, it seems that signalling to a multilingual language model what the next language will be through its initial state is a good idea, and that by doing through the means of a vector containing language features bears no disadvantage, and can be positively helpful when having the model learn a new language.

6.2. Future Work

In future work, it would be interesting to re-examine the WALS-enhanced sequence to sequence model in the context of multilingual translation, as it might make different use of the WALS data than the language model. Furthermore, by developing a fully-fledged translation model, information from WALS could not only be used in the encoder, but also the decoder, by for example creating special WALS-enhanced tokens to start decoding.

Another possibility would be to pre-train a language/translation model with specialized data that would more explicitly link the information contained in the WALS vectors with their true linguistic meanings. For example making sure the model understands the difference between the adjective-noun and the noun-adjective order in sentences of different languages, and recognizes where this information can be found in the WALS vector. The difference in behaviour on such a grammatical level could also potentially be of use in systems which use part of speech tags, as the connections there could be made more explicitly.

Bibliography

[1] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. 19, 2 (June 1993), 263-311. 4

[2] Dryer, Matthew S. & Haspelmath, Martin (eds.) 2013. The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. (Available online at `http://wals.info`, Accessed on 15/06/2017.) 13, 14

[3] WIT^3 – Web Inventory of Transcribed and Translated Talks `https://wit3.fbk.eu/` (Accessed on 08/09/2017) 17

[4] M. Cettolo, C. Girardi, and M. Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. In Proc. of EAMT, pp. 261-268, Trento, Italy. 17

[5] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735. 7

[6] Christopher Olah, Understanding LSTM Networks, Aug.2015. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/` (Accessed on 03/09/2017) 7

[7] Kjell Magne Fauske, Example: Neural network `http://www.texample.net/tikz/examples/neural-network/` Dec. 2006. (Accessed on 08/09/2017) 6

[8] GitHub - JayParks/tf-seq2seq: Sequence to sequence (seq2seq) learning Using TensorFlow. `https://github.com/JayParks/tf-seq2seq` (Accessed on 10/08/2017) 17

[9] Tensorflow: Sequence-to-Sequence Models `https://www.tensorflow.org/tutorials/seq2seq` (Accessed on 05/08/2017) 17

[10] G. Neubig. Neural Machine Translation and Sequence-to-sequence Models: A Tutorial. ArXiv e-prints, Mar. 2017. 5, 7, 9

[11] GitHub - rsennrich/subword-nmt: Subword Neural Machine Translation `https://github.com/rsennrich/subword-nmt` (Accessed on 23/06/2017) 9

[12] Rico Sennrich, Barry Haddow and Alexandra Birch (2016): Neural Machine Translation of Rare Words with Subword Units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). Berlin, Germany. 9, 11

[13] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1988. Learning representations by back-propagating errors. In Neurocomputing: foundations of research, James A. Anderson and Edward Rosenfeld (Eds.). MIT Press, Cambridge, MA, USA 696-699. 6

[14] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi and Jürgen Schmidhuber. 2001. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies 7

[15] K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. Neural Netw. 2, 5 (July 1989), 359-366. `http://dx.doi.org/10.1016/0893-6080(89)90020-8` 5

[16] Yoav Goldberg. A primer on neural network models for natural language processing. ArXiv e-prints, arXiv:1510.00726, Oct. 2015. 6

[17] Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ArXiv e-prints, arXiv:1409.0473, Sep. 2014.

[18] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. ArXiv e-prints, arXiv:1611.04558, Nov. 2016. 2, 11

[19] Orhan Firat, Kyunghyun Cho and Yoshua Bengio. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. ArXiv e-prints, arXiv:1601.01073, Jan. 2016. 2, 11

[20] Thanh-Le Ha, Jan Niehues, Alexander Waibel. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. ArXiv e-prints, arXiv:1611.04798, Nov. 2016. 2, 11

[21] Alexander Waibel, Toshiyuki Hanazawa, Geofrey Hinton, Kiyohiro Shikano, and Kevin J. Lang. 1990. Phoneme recognition using time-delay neural networks. In Readings in speech recognition, Alex Waibel and Kai-Fu Lee (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA 393-404. 5

[22] Minh-Thang Luong and Christopher D. Manning. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. ArXiv e-prints, arXiv:1604.00788, Apr. 2016. 11

[23] Junyoung Chung, Kyunghyun Cho and Yoshua Bengio. A Character-Level Decoder without Explicit Segmentation for Neural Machine Translation. ArXiv e-prints, arXiv:1603.06147, Mar. 2016. 11

[24] Wang Ling, Isabel Trancoso, Chris Dyerand Alan W. Black. Character-based Neural Machine Translation. ArXiv e-prints, arXiv:1511.04586, Nov. 2015. 11

[25] Yoon Kim, Yacine Jernite, David Sontag and Alexander M. Rush. Character-Aware Neural Language Models. ArXiv e-prints, arXiv:1508.06615, Aug. 2015. 11

[26] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer and Yonghui Wu. Exploring the Limits of Language Modeling. ArXiv e-prints, arXiv:1602.02410, Feb. 2015. 11

[27] Tomas Mikolov, Quoc V. Le and Ilya Sutskever. Exploiting Similarities among Languages for Machine Translation. ArXiv e-prints, arXiv:1309.4168, Sep. 2013. 11

[28] Lauriane Aufrant, Guillaume Wisniewski and François Yvon. Zero-resource Dependency Parsing: Boosting Delexicalized Cross-lingual Transfer with Linguistic Knowledge. COLING. Dec. 2016. 11

[29] Joachim Daiber, Miloš Stanojević and Khalil Sima'an. Universal Reordering via Linguistic Typology. COLING. Dec. 2016. 11

[30] Östling, Robert. Word Order Typology through Multilingual Word Alignment. 2. 205-211. 10.3115/v1/P15-2034. 2015. 11

[31] Arianna Bisazza, Marcello Federico. A Survey of Word Reordering in Statistical Machine Translation: Computational Models and Language Phenomena. ArXiv e-prints, arXiv:1502.04938, Feb. 2015. 11

Appendices

A. Sample Outputs from Multilingual Character Based Language Models

Presented below are sample outputs from the two 13 language multilingual character based language models trained during the experiments.

Language	Sample Text (WALS)	Language	Sample Text (IDs)
Czech	děkuji.	Czech	děkuji.
German	und das ist eine andere.	German	vielen dank.
English	thank you.	English	thank you.
Spanish	así que esto es lo que hacemos.	Spanish	esto es lo que hacen.
French	merci.	French	merci.
Croatian	to je ono što možemo.	Croatian	to je bilo N.
Hungarian	all.	Hungarian	ok.
Italian	ok.	Italian	no.
Dutch	dank u.	Dutch	dit is een voorbeeld.
Polish	dziękuję.	Polish	dziękuję.
Portuguese	o que é que estamos a fazer?	Portuguese	obrigado.
Romanian	de ce?	Romanian	de ce?
Russian	это не так.	Russian	спасибо .
Turkish	teşekkürler.	Turkish	teşekkürler.
Ukrainian	дякую.	Ukrainian	дякую.

Table A.1.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 1.

Language	Sample Text (WALS)	Language	Sample Text (IDs)
Czech	děkuji vám.	Czech	děkuji vám.
German	und das ist eine andere art.	German	und das ist eine art.
English	thank you very much.	English	thank you very much.
Spanish	así que eso es lo que hacemos.	Spanish	esto es lo que hace.
French	merci beaucoup.	French	merci beaucoup.
Croatian	to je ono što možemo.	Croatian	to je bilo dobro.
Hungarian	multiple N.	Hungarian	ok?
Italian	ok.	Italian	dank.
Dutch	dank je.	Dutch	dit is een beetje.
Polish	dziękuję bardzo.	Polish	dziękuję bardzo.
Portuguese	o que é que estamos a fazer?	Portuguese	obrigada.
Romanian	de ce?	Romanian	de ce?
Russian	это просто.	Russian	спасибо!
Turkish	dedi.	Turkish	tamam.
Ukrainian	дякую вам.	Ukrainian	дякую!

Table A.2.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 2.

Language	Sample Text (WALS)
Czech	proč?
German	und das ist eine andere art.
English	thank you very.
Spanish	así que esto es lo que estamos haciendo.
French	vous voyez?
Croatian	to je ono što možemo.
Hungarian	all.
Italian	ok.
Dutch	dank u.
Polish	to jest niesamowite.
Portuguese	o que é que estamos a fazer?
Romanian	de ce?
Russian	это не просто.
Turkish	tamam.
Ukrainian	це не так.

Language	Sample Text (IDs)
Czech	takže jsem se stalo.
German	und das ist eine art.
English	thank you very much!
Spanish	esto es lo que hacen.
French	nous avons besoin de la vie.
Croatian	to je bilo N.
Hungarian	ok?
Italian	dank.
Dutch	dit is een beetje.
Polish	dziękuję bardziej.
Portuguese	porquê?
Romanian	de ce?
Russian	спасибо!
Turkish	dedi.
Ukrainian	дякую!

Table A.3.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones. Beam 3.

Language	Sample Text (WALS)
Czech	je to tak?
German	und das ist eine andere art.
English	thank you very.
Spanish	así que eso es lo que estamos haciendo.
French	vous voyez?
Croatian	to je ono što mislite.
Hungarian	all.
Italian	então , o que aconteceu?
Dutch	het is een beetje.
Polish	to jest niesamowita.
Portuguese	portanto , podemos fazer isso.
Romanian	de ce?
Russian	это просто.
Turkish	teşekkür ederim.
Ukrainian	я не знаю.

Language	Sample Text (IDs)
Czech	takže jsem se stalo.
German	und das ist eine art.
English	that's what we do.
Spanish	esto es lo que hace.
French	nous avons besoin de la maison.
Croatian	to je bilo jedno.
Hungarian	ok?
Italian	alternativa.
Dutch	dit is een voorbeeld van de mensen.
Polish	dziękuję bardziej.
Portuguese	porquê?
Romanian	de ce?
Russian	я не знаю.
Turkish	teşekkür ederim.
Ukrainian	дякую за увагу.

Table A.4.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones. Beam 4.

Language	Sample Text (WALS)
Czech	děkuji vám. "
German	aber das ist eine andere menschen.
English	thank you very.
Spanish	así que esto es lo que estamos haciendo esto.
French	vous voyez?
Croatian	to je ono što mislite.
Hungarian	multiple N.
Italian	então , o que estamos a construir uma estratégia de comunicação.
Dutch	dank je.
Polish	powiedział : " nie.
Portuguese	o que é que estamos a fazer?
Romanian	de ce?
Russian	и это просто.
Turkish	teşekkürler?
Ukrainian	я не знаю.

Language	Sample Text (IDs)
Czech	takže jsem se stalo na to.
German	und das ist eine großartige design.
English	thank you very much!
Spanish	esto es lo que hace.
French	nous avons commencé à la maison.
Croatian	to je bilo jednostavna.
Hungarian	design.
Italian	no.
Dutch	dit is een beetje.
Polish	dziękuję bardzo , ale nie wiedzieliśmy , że jest to niesamowite.
Portuguese	porque é que estamos a falar?
Romanian	de ce?
Russian	я не знаю.
Turkish	diye sordu.
Ukrainian	але це не так.

Table A.5.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 5.

Language	Sample Text (WALS)
Czech	proč?
German	aber das ist eine andere menschen.
English	thank you very different.
Spanish	así que eso es lo que estamos haciendo.
French	c'est ce que nous avons fait.
Croatian	to je ono što se dogodi.
Hungarian	muchas generacion.
Italian	então , o que estamos a construir uma estratégia de comunicação.
Dutch	het is een andere probleem.
Polish	powiedziałem : " nie.
Portuguese	porque é que estamos a fazer?
Romanian	acesta este un exemplu despre aceasta.
Russian	это просто.
Turkish	teşekkürler!
Ukrainian	це було просто.

Language	Sample Text (IDs)
Czech	takže jsem se stalo na to , co se stalo.
German	und das ist eine art.
English	that's what we do.
Spanish	esto es lo que hacer.
French	nous avons besoin de la main.
Croatian	to je bilo jedno.
Hungarian	new york city.
Italian	disponibility.
Dutch	dit is een voorbeeld is.
Polish	dziękuję bardziej.
Portuguese	porque é que estamos a falar?
Romanian	acesta este un problem.
Russian	спасибо!
Turkish	diye sordu.
Ukrainian	дякую!

Table A.6.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 6.

Language	Sample Text (WALS)
Czech	proč?
German	und das ist eine andere geschichte ist.
English	thank you very simple.
Spanish	así que esto es lo que estamos haciendo en el mundo.
French	vous pouvez voir quelque chose.
Croatian	to je ono što mislite.
Hungarian	muchas generativa.
Italian	então , o que estamos a construir uma coisa que estamos a construir uma coisa.
Dutch	het is een andere probleem.
Polish	powiedział : " nie.
Portuguese	porque é que estamos a fazer?
Romanian	acesta este un moment.
Russian	это не только просто.
Turkish	teşekkürler?
Ukrainian	це було просто.

Language	Sample Text (IDs)
Czech	takže jsem se stalo na to.
German	aber das ist eine großartige design.
English	thank you very much , you know.
Spanish	esto es lo que estamos hacen.
French	nous avons besoin de la main.
Croatian	to je bilo jednostavno , ali ne možemo učiniti.
Hungarian	ok?
Italian	alternativa est total.
Dutch	dit is een beetje.
Polish	dziękuję bardzo , ale nie wiedzieliśmy , że jest to prawda.
Portuguese	porque é que estamos a fazer? "
Romanian	acesta este un moment de aceasta.
Russian	спасибо больше.
Turkish	diye sordum.
Ukrainian	вони не знають.

Table A.7.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 7.

Language	Sample Text (WALS)
Czech	takže to je to , co se stalo.
German	und das ist eine andere geschichte , die ich diese menschen auf diesem moment.
English	thank you very much of.
Spanish	así que esto es lo que estamos haciendo algunas cosas.
French	c'est ce que nous avons fait.
Croatian	to je ono što možemo napraviti nešto.
Hungarian	muchas generation.
Italian	então , o que estamos a construir uma coisa que estamos a construir uma escola.
Dutch	het is een andere manier van de mensen.
Polish	powiedziałem : " nie...
Portuguese	porque é que estamos a fazer isto?
Romanian	acesta este un exemplu despre cel mai mult.
Russian	это очень просто , который мы начинаем.
Turkish	teşekkürler....
Ukrainian	я не знаю , що це було не так.

Language	Sample Text (IDs)
Czech	takže jsem se stalo na to , co se stalo , že je to velmi dobré.
German	aber das ist eine großartige design.
English	that's what we do.
Spanish	esto es lo que hacer.
French	nous avons besoin de la construction de la maison.
Croatian	to je bilo jednostavno , ali ne možemo učiniti da je to napravio.
Hungarian	new york city.
Italian	alternativa , design , design.
Dutch	dit is een voorbeeld is.
Polish	dziękuję bardzo , że nie wiedziałam.
Portuguese	porquê?
Romanian	acesta este un moment de această persoană.
Russian	спасибо больше.
Turkish	teşekkürler!
Ukrainian	вони не знають.

Table A.8.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 8.

Language	Sample Text (WALS)
Czech	takže to je to , co se stane , že jsem se podíval na to , že jsem se na to podíval.
German	aber das ist eine andere geschichte ist.
English	thank you very much of.
Spanish	así que esto es lo que estamos haciendo algunas personas que están en el mundo.
French	c'est ce que nous avons fait.
Croatian	to je ono što možemo napraviti nešto.
Hungarian	muchas generatives.
Italian	então , o que estamos a construir uma coisa que estamos a construir uma coisa que estamos a trabalhar para as pessoas que estamos a trabalhar para os mesmos.
Dutch	het is een andere manier van de mensen.
Polish	powiedziałem : " nie...
Portuguese	portanto , o que é que estamos a fazer?
Romanian	acesta este un exemplu despre cel mai mare.
Russian	это очень просто , который мы начинаем.
Turkish	teşekkürler....
Ukrainian	я не знаю , що це було не так.

Language	Sample Text (IDs)
Czech	takže jsem se stalo na to , co se stalo , že je to velmi dobrý.
German	und das ist eine großartige versuche.
English	thank you very much , you know , that's what we do it.
Spanish	esto es lo que hacer.
French	nous avons besoin de la main.
Croatian	to je bilo jednostavno , ali ne možemo učiniti da je to napraviti.
Hungarian	new york times don't simple personne.
Italian	alternativa , design , designer.
Dutch	dit is een voorbeeld van de verschillende staten.
Polish	dziękuję bardzo , że nie wiedziałam.
Portuguese	porque é que estamos a fazer isso?
Romanian	acesta este un moment de această persoană a fost într-un fel de mare.
Russian	я не знаю , что это не так.
Turkish	teşekkürler!
Ukrainian	але це не так?

Table A.9.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 9.

Language	Sample Text (WALS)
Czech	takže to je to , co se stane , že jsem se podíval na to , že jsem se na to představili.
German	und das ist eine andere geschichte , die ich diese menschen auf diesem moment.
English	thank you very much of.
Spanish	así que esto es lo que estamos haciendo algunas personas que están en el mundo de la compañía.
French	vous pouvez voir quelque chose qui se passe dans le monde , et c'est une compassion.
Croatian	to je ono što možemo napraviti nešto što se događa.
Hungarian	muchas generatives.
Italian	então , o que estamos a construir uma coisa que estamos a construir uma coisa que estamos a trabalhar para as pessoas que estamos a trabalhar para os problemas para os mesmos destas partes de comportamento.
Dutch	het is een andere manier van de mensen.
Polish	powiedziałem : " nie...
Portuguese	portanto , o que é que acontece?
Romanian	acesta este un moment.
Russian	это очень просто , который мы не знаем.
Turkish	teşekkürler!
Ukrainian	я не знаю , що це було не просто не завжди.

Language	Sample Text (IDs)
Czech	takže jsem se stalo na to , co se stalo se stalo , aby se stalo se stalo.
German	und das ist eine großartige geschichte , die ich die geschichte , die ich eine andere menschen auf diesem grund.
English	thank you very much , you know , that's what i was a little bit.
Spanish	esto es lo que estamos haciendo algunos de estas cosas que se puede hacer esto.
French	nous avons besoin.
Croatian	to je bilo jednostavno , ali ne možemo učiniti da je to napraviti na svijet.
Hungarian	new york times don't simple personnal.
Italian	alternativa , design , design , a design.
Dutch	dit is een voorbeeld van de verschillende staten.
Polish	dziękuję bardzo , ale nie wiedzieliśmy , że jest to prawda?
Portuguese	porque é que estamos a acontecer? "
Romanian	acesta este un moment de această persoană a fost într-un fel de mult de asta.
Russian	спасибо большое , что вы видите , что они не просто не понимают , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите , что вы видите?
Turkish	teşekkür ederim , ama bunu yapmak için bir şey.
Ukrainian	але це не так?

Table A.10.: Sample output from the multilingual language models for the 13 learned languages and 2 unlearned ones . Beam 10.