CHAPTER 7

*Quantifying the Effects of Multilinguality in NLP Sequence Prediction Tasks

Abstract|The fact that languages tend to share certain properties can be exploited by, e.g., sharing parameters between languages. This type of model multilinguality is relatively common, as taking advantage of language similarities can be beneficial. However, the question of *when* multilinguality is a useful addition in terms of monolingual model performance is left unanswered. In this chapter, we explore this issue by experimenting with a sample of 60 languages on a selection of tasks: semantic tagging, part-of-speech tagging, dependency relation tagging, and morphological inflection. We compare results under various multilingual model transfer conditions, and finally observe correlations between model effectivity and two measures of language similarity.

*Chapter adapted from: **Bjerva, J.** (in review) Quantifying the Effects of Multilinguality in NLP Sequence Prediction Tasks.

7.1 Introduction

Languages tend to resemble each other on various levels, for instance by sharing syntactic, morphological, or lexical features. Such similarities can have many different causes, such as common language ancestry, loan words, or being a result of universals and constraints in the properties of natural language itself (cf. Chomsky (2005); Hauser et al. (2002)). Several current approaches to problems in Natural Language Processing (NLP) take advantage of these similarities. For instance, in model-transfer settings, parsers are frequently trained on (delexicalised) versions of entire treebanks, whereas in annotation projection, word alignments between translated sentences are used (McDonald et al., 2011b; Täckström et al., 2012; Tiedemann, 2015; Ammar et al., 2016; Vilares et al., 2016; Agić et al., 2016).[2] In the case of language modelling, multilinguality can be taken advantage of, e.g., in order to model domain-specific or diachronically specific language variants (Östling and Tiedemann, 2017). In addition to these specific examples, multilinguality in NLP models has been used in a whole host of other tasks, such as part-of-speech (PoS) tagging and semantic textual similarity, and is especially useful for NLP for low resource languages (Georgi et al., 2010; Täckström et al., 2013; Faruqui and Lample, 2016; Agić et al., 2017). One concrete advantage of taking a multilingual approach, is that this allows for the exploitation of much larger amounts of data, as compared to using monolingual approaches. This fact, together with the prevalence of multilingual approaches in modern NLP, highlight the importance of further research in this area.

Although the literature contains a large amount of successful multilingual approaches, it is not sufficiently clear in which cases multilinguality is likely to be helpful. The previous chapter served as an example for this, with some tentative indications of which com-

[2] These approaches are covered in Chapter 3.

binations were useful, based on typological relatedness. Hence, it may be reasonable to assume that choosing typologically similar languages when building a multilingual model will be beneficial, however this is not always the case, and relying on intuition or personal language knowledge in such matters has its limitations. Hence, the go-to approach when considering multilinguality as a means of performance improvement in an NLP setting, is the time-consuming and resource-exhausting process of trial and error. In this chapter, the aim is to provide insight into how one might approach the selection of languages when considering model multilinguality. We investigate the following research questions, in order to provide an answer to **RQ 4**:

RQ 4a Given a model trained on a language, l_1, does adding data for another language, l_2, increase the performance on l_1 if those languages are similar?

RQ 4b In which way can such similarities be quantified?

RQ 4c What correlations can we find between model performance and language similarities?

We experiment on four NLP tasks: semantic tagging, part-of-speech tagging, dependency relation tagging, and morphological inflection. The language sample we use differs per task, and covers a total of 60 languages from a typologically diverse sample. After covering related work, we first present experiments in multilingual settings for each of these tasks (Sections 7.2, 7.3, and 7.4). We then investigate the correlations between two different measures of language similarity, and the change in system performance observed in multilingual settings, in order to provide an answer to our research questions (Section 7.5). An approach which can be considered as parallel to this effort, is works similar to Rosa et al. (2017), in which sim-

ilarities between languages are exploited when deciding on which features to use in a cross-lingual parser.

7.2 Semantic Tagging

7.2.1 Background

The first task under consideration is semantic tagging, as introduced in Bjerva et al. (2016b), and described in more detail in Chapter 4. In this chapter, we consider this task in a multilingual setting, which is possible since the Parallel Meaning Bank (PMB, Abzianidze et al., 2017) includes such data for four languages: English, Dutch, German, and Italian.

7.2.2 Data

We use semantic tagging data obtained from the PMB (Abzianidze et al., 2017). There is a relatively large amount of gold standard annotation for English, which we use in our experiments. Note that, we only use data from the PMB in this setting, as opposed to the setting in Chapter 4 in which we also use data from the Groningen Meaning Bank (GMB, Bos et al., 2017). For the languages other than English, i.e., Dutch, German, and Italian, we rely on the projected tags based on this gold standard annotation. These tags were projected as described by Abzianidze et al. (2017), using word alignments obtained with *GIZA++* (Och and Ney, 2003). As the amount of parallel text differs per language, this yields the data amounts listed in Table 7.1.

7.2.3 Method

We employ a relatively simple neural network tagger for all of the tagging tasks in this study. The tagger used is a bi-directional Long Short-Term Memory model (Bi-LSTM, Hochreiter and Schmidhuber (1997); Graves and Schmidhuber (2005)). We use a single hid-

Table 7.1: Overview of the semantic tagging data used in this work.

Language	Tokens	Sentences	Status
English	20,098	2,814	Gold
Dutch	3,446	506	Projected
German	13,702	1,960	Projected
Italian	11,376	1,711	Projected

den layer for each direction, as shown in Figure 7.1. The input-representations used are 100-dimensional multilingual word embeddings trained on UN (Ziemski et al., 2016), Europarl (Koehn, 2005), and Bible data, using multilingual skip-gram (Guo et al., 2016), based on word alignments obtained with a variant of EFMARAL (Östling and Tiedemann, 2016).[3,4]

The neural architecture used in this experiment is simpler than, e.g., the deep residual network Bi-GRU presented in Chapter 4. This choice was made mainly for three reasons. The primary motivation is that we wanted to rely only on multilingual word representations, seeing how far this will get us, without dealing with morphological dissimilarities, or exploiting morphological similarities directly. Additionally, using only word representations is one way of using fewer parameters, meaning that fewer computational resources are needed. Finally, although systems using character-based representations generally perform better than ones using only word-based representations, we are not interested in absolute performance *per se*, but rather relative changes in performance when building multilingual models.

[3]Using the default parameter settings for *eflomal*: `https://github.com/robertostling/eflomal`.

[4]These are the same embeddings used in the previous chapter.

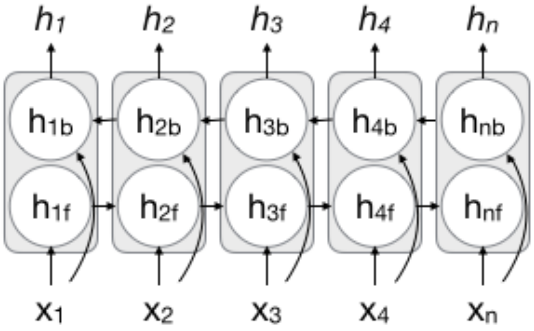


Figure 7.1: Sketch of the Bi-LSTM architecture used for our tagging tasks.

Table 7.2: Hyperparameters used for semantic tagging

Hyperparameter	**Setting**	**Notes**
Library	Chainer (Tokui et al., 2015)	
Loss function	Categorical Cross-Entropy	
Optimiser	Adam (Kingma and Ba, 2014)	
Training iterations	Early stopping, *best val loss*	
Batch size	4 sentences	
Regularisation	Dropout (Srivastava et al., 2014)	$p = 0.5$
Regularisation	Weight decay (Krogh and Hertz, 1992)	$\epsilon = 10^{-4}$

Hyperparameters

We use the same hyperparameter settings for each of the experimental settings, so as to ensure comparability. These are detailed in Table 7.2. We always train for a maximum of 50 epochs, using the epoch at which validation loss was the best for evaluation.

7.2.4 Experiments and Analysis

We first look at a high-resource to low-resource scenario, using the languages for which we have a large amount of data (English, Ger-

man, and Italian) as source languages, and all four languages as target languages. We then run further experiments using all four languages as source languages. Since only relatively few tokens are available for Dutch we train on 1000 tokens, as this allows us to compare all four languages equally. We reserve 500 tokens for each language as test data, and split the remaining data into 80% for training and 20% for validation. Since we deal with parallel texts, we make sure that the training, development, and test sets used for l_1 and any l_2 do not overlap in any way during training.

Transfer from high-resource to low-resource languages

The goal of this experiment is to investigate zero-shot learning between languages for semantic tagging, in a scenario in which we use all training data available, giving us access to between 10,000 and 20,000 tokens of annotated data for the source languages. Given the tentative results of the previous chapter, we expect that the Germanic languages will be more beneficial for one another, as compared to Italian.

Table 7.3: Results on high-resource to low-resource transfer. Bold indicates the best source language for each target language, not considering the cases in which the source and target languages are identical, which are denoted by italics.

Test \ Train	English	German	Italian
English	*75.03%*	**49.20%**	35.45%
Dutch	49.30%	**56.90%**	36.31%
German	**41.99%**	*67.41%*	41.17%
Italian	35.74%	**39.11%**	*70.89%*

The results from these experiments are shown in Table 7.3, in which bold represent the best source languages for each target lan-

guage, and italics denote the cases in which source and target languages are the same. We can observe that the results when the source and target languages are both Germanic are higher than when the Romance language Italian is involved. For instance, using German as a source language for Dutch results in relatively high accuracies. This can be explained by two factors. For one, it is likely that the quality of the multilingual word embeddings is higher when comparing German and Dutch. Additionally, the extensive similarities between these two languages is likely to make zero-shot learning relatively easy.

Note that these results are not strictly comparable to those in Chapter 4, since we both use different and less training and evaluation data. In absolute terms, performance on similar data is around 5% worse in this experimental setting, although we train on approximately one order of magnitude less data than in Chapter 4.[5]

Transfer from low-resource source languages

In the low-resource source scenario, with 1000 training tokens per language, we compare some linguistic and input representation conditions. We run experiments with two *input representation* settings: i) with *frozen* pre-trained word representations; ii) with *updated* pre-trained word representations. By *frozen* word representations, we refer to representations which are kept at their initial states during learning. That is to say, errors are not back-propagated into the embeddings. In the *updated* condition, however, embeddings are updated during training. This comparison is done so as to investigate whether there is a difference when enforcing the multilingual embedding space to remain in its initial state, thus preserving multilingual distances.

[5] In order to compare the architectures used, we also train and evaluate our tagger on the same data as in Chapter 4, in which case we obtain approximately the same performance as the baseline using only word representations.

In combination with the two representation settings, we also use two *linguistic* settings: a) monolingual training; and b) multilingual training. The monolingual training serves as a baseline, indicating how well we can transfer semantic tags from a source language to a target language, by only training on the target language. In the multilingual setting, we add training data for the target language, comparing transfer between languages in a multilingual setting. This comparison is done so as to investigate to what extent we can take advantage of data from two low resource languages, with the goal of benefitting both languages.

We first present results from the monolingual training in both input-representation settings, in Tables 7.4a, and 7.4b.

Table 7.4: Results on monolingual semantic tagging.

(a) Training on 1k monolingual tokens, frozen embeddings

Test \ Train	English	Dutch	German	Italian
English	**64.54%**	37.24%	36.40%	28.54%
Dutch	36.32%	**53.20%**	44.80%	28.75%
German	35.10%	39.12%	**54.36%**	37.31%
Italian	30.29%	30.43%	29.40%	**61.20%**

(b) Training on 1k monolingual tokens, updated embeddings

Test \ Train	English	Dutch	German	Italian
English	**64.90%**	38.82%	37.43%	31.51%
Dutch	39.86%	**56.78%**	40.93%	30.81%
German	39.14%	40.23%	**55.46%**	39.07%
Italian	25.09%	34.02%	31.21%	**49.63%**

Not surprising, monolingual models trained on the target language consistently perform better than when the source language is

different from the target language, as in zero-shot learning. Nonetheless, all results when source and target languages differ outperform a most frequent class baseline (17.35%) by far. This is expected, as the pre-trained models have been trained on a relatively large amount of parallel data, and have formed word spaces which are unified across languages, which allows them to generalise somewhat across languages, and confirms the quality of the embeddings themselves. Comparing these results to those of the high-resource to low-resource setting, we can observe a steep drop in performance. This is due to the fact that we have approximately an order of magnitude less data in the current setting.

Surprisingly, updating the pre-trained word embeddings during training increases results for most source/target combinations (English/Italian being the exception). This was unexpected, as it is usually beneficial to update such embeddings during training, as this allows them to learn representations which are tuned for the task at hand. It was, however, not expected that this would be the case when applying model transfer, as we expected that tuning, say, the English representation for *dog* to be more task-specific, would skew it away from that of the Dutch equivalent *hond*. It would be interesting to explore this further, observing the resulting word-space after updating only one language in a multilingual language space. Italian, interestingly, sees a severe drop in performance when updating embeddings in a monolingual setting. This might be explained by the updated embeddings being overfit, and not generalising to the test set.

Transfer between low-resource source languages

We now turn to transfer between low-resource source languages. In this setting, we are mainly interested in seeing whether more related languages, i.e. English, Dutch, and German, are more beneficial to

combine with one another, than with the typologically more distant Italian language.[6]

Table 7.5: Results on multilingual semantic tagging.

(a) Training on 1k+1k multilingual tokens, frozen embeddings

Test \ Train	English	Dutch	German	Italian
English	64.54%	**65.38%**	64.98%	65.08%
Dutch	56.07%	53.20%	**60.20%**	54.96%
German	54.96%	**55.30%**	54.36%	54.49%
Italian	47.69%	47.25%	47.29%	**61.20%**

(b) Training on 1k+1k multilingual tokens, updated embeddings

Test \ Train	English	Dutch	German	Italian
English	64.90%	39.53%	**64.98%**	22.85%
Dutch	37.56%	56.78%	**60.20%**	26.69%
German	39.21%	40.12%	**55.46%**	21.87%
Italian	26.27%	34.47%	47.29%	**49.63%**

Tables 7.5a, and 7.5b contain the results from semantic tagging with multilingual training. Considering the rows in each table, it is generally the case that training a model with a combination of English, Dutch and German, improves results more than combining one of these with Italian. This seems to hold in both conditions, with and without updating the pre-trained vectors in training.

In contrast to the monolingual training case, we here do observe that freezing the vectors during training is beneficial for model performance. It may be the case that, since the weights of the embeddings in both l_1 and l_2 are optimised, they are pushed even further apart than in the monolingual training case in which only one lan-

[6]We will consider how these similarities can be quantified in Section 7.5.

guage's embeddings are affected. Hence, in the frozen case, the integrity of the multilingual language space is maintained, allowing the model to learn cross-lingually. A potential explanation for the drop in results on Italian when updating the embeddings might be the extent to which the languages are similar. Since English, German, and Dutch are all Germanic languages, it is possible that this relatedness suffices to preserve the multilingual quality of the word space.

7.2.5 Summary of Results on Semantic Tagging

We have observed that training on similar languages is helpful for semantic tagging, in the sense that combining Germanic languages tended to be beneficial. Additionally, training in a high-resource scenario on, e.g., German and using Dutch as a source language yielded better results than when training on low-resource Dutch (see Tables 7.3 and 7.4a). This leads us to ask whether similar patterns can be found when observing a larger sample of languages, and on other tasks.

7.3 Tagging Tasks in the Universal Dependencies

The Universal Dependencies treebank offers an excellent testing ground for experiments on NLP model multilinguality. The corpus collection contains many languages, with several layers of uniform annotation across languages (Nivre et al., 2016b). We use version 2.0 of the UD treebanks for experiments in two tasks: PoS tagging and dependency relation tagging (Nivre et al., 2017). We evaluate on the 48 languages for which training data is available.

7.3.1 Data

In order to balance our experiments for differing data sizes, we balance all training sets so as to have an equal number of tokens. We

set this amount to 20,000 tokens, in order to allow for inclusion of the smallest language in the UD (Vietnamese, $n = 20285$). Hence, the overall results obtained will be relatively low, but should make the effects of multilingual modelling clearer. A part of the evaluation will deal with grouping languages per language group. An overview of which languages are included in the Germanic, Romance, and Slavic families in these evaluations is given in Table 7.6.

7.3.2 Method

We employ the same tagger as described in the semantic tagging experiments of this chapter, detailed in Section 7.2.3. We also use the same hyperparameter settings, as shown in Table 7.2, and the same input representations. The main difference with the semantic tagging experiments is therefore simply the tasks at hand, and the amount of languages under consideration.

Experimental Setup

We only use the setting with *frozen* word representations, as established in the semantic tagging experiments. We focus on results from the multilingual training settings, as we are interested in how these results differ between language pairs. Additionally, we consider two tasks in this setup: PoS tagging, and dependency relation tagging. Dependency relation tagging is the task of predicting the dependency tag (and its direction) for a given token. This is a task that has not received much attention, although it has been shown to be a useful feature for parsing (Ouchi et al., 2014, 2016).[7] These *deprel* tags can be derived directly from UD dependency parse trees, making it straight-forward to evaluate on this task for the same sample of languages in the same settings. In this setting, we use the dependency

[7] Dependency relation labels are discussed in more detail in Chapter 5.

Table 7.6: Language grouping of the Germanic, Romance, and Slavic languages used in our experiments.

Language family	Language
Germanic	Afrikaans Danish Dutch English German Norwegian Bokmål Norwegian Nynorsk Swedish
Slavic	Belarusian Bulgarian Croatian Czech Old Church Slavonic Polish Russian Serbian Slovak Slovenian Ukrainian
Romance	Catalan French Galician Italian Latin Portuguese Brazilian Portuguese Romanian Spanish

relation instantiations with simple granularity and simple directionality (i.e., encoding the head and its relative position, for each word), described further in Chapter 5 (Table 5.1).

7.3.3 Results and Analysis

Due to the large number of language pairs, we discuss the results from the mean accuracy on a language group when trained in combination with a sample of languages.[8]

PoS Tagging

Table 7.7 contains PoS tagging results with frozen embeddings. Some noteworthy findings include the highest accuracies per language group, marked in bold. These are generally obtained by languages which are in the same language group, although there are exceptions to this pattern. Note that we do not develop on the language which we use for evaluation. That is to say, e.g., when evaluating on Danish, the *Germanic* column is calculated as the mean accuracy over German, Norwegian Bokmål and Nynorsk, and not including Danish.

There are examples in which training on a language from the same language group worsens performance overall. Some notable cases include training on Dutch for the Germanic languages. The relatively poor performance as compared to Danish might be explained by the fact that this group includes four Scandinavian languages, meaning that Danish has three such languages which it is likely helpful for. Dutch, on the other hand, has only two languages to which it is highly similar in the Germanic group, namely Afrikaans and German.

It is nonetheless somewhat puzzling that some non-Germanic languages yield better performance in the Germanic group than Dutch.

[8]Results covering all languages are presented later in this chapter.

Considering the baseline column in the table, however, it is clear that the model only sees increases in performance in a few cases, with a loss in performance in nearly all cases. Therefore a potential explanation to the overall drop in results might be the fact that, although all languages within a single group are related to one another, this relatedness might still be too distant to be exploited in the current setup. For instance, the languages in the Slavic group represent a relatively large variety.

Table 7.7: PoS results – Training on 20k+20k multilingual tokens, frozen embeddings. Columns indicate average results over languages in that language group.

Language	**Germanic**	**Romance**	**Slavic**
Baseline	83.97%	84.35%	81.12%
Bulgarian	83.89%	82.93%	76.53%
Czech	83.79%	82.34%	76.28%
Danish	**84.14%**	84.20%	79.73%
Finnish	83.87%	83.58%	78.41%
French	82.59%	**84.54%**	79.73%
Italian	83.83%	83.77%	79.67%
Dutch	82.46%	84.32%	79.40%
Polish	81.93%	84.44%	78.58%
Portuguese	83.35%	81.14%	78.42%
Russian	83.85%	82.77%	**82.26%**

Dependency Relation Tagging

Table 7.8 contains results from dependency relation tagging in the frozen embeddings setting. Interestingly, although the top results for Germanic and Slavic are from in-group languages, we observe the best results here for out-of-group languages for the Romance group. The results of these experiments show a similar trend to those in

the PoS experiments, with almost all results being worse than the baseline.

Table 7.8: DepRel results – Training on 20k+20k multilingual tokens, frozen embeddings. Columns indicate average results over languages in that language group.

Language	Germanic	Romance	Slavic
Baseline	65.10%	70.31%	65.25%
Bulgarian	59.01%	**69.52%**	60.31%
Czech	64.05%	67.91%	61.95%
Danish	**65.56%**	68.20%	61.74%
Finnish	64.87%	63.34%	61.78%
French	64.99%	67.63%	62.23%
Italian	63.94%	67.97%	61.91%
Dutch	64.71%	66.33%	62.29%
Polish	64.15%	64.68%	60.98%
Portuguese	64.92%	67.36%	62.31%
Russian	64.17%	68.32%	**65.30%**

7.3.4 Summary of Results on the Universal Dependencies

For semantic tagging, we saw increases in performance when combining the Germanic languages with one another. The results from tagging tasks on the UD languages reveal that this granularity of language similarity is not sufficient to determine whether this type of model multilinguality will be successful, under the experimental conditions used here. In fact, observing results aggregated by the language families Germanic, Romance, and Slavic, revealed a decrease in performance when transferring from almost all languages in these families, with some exceptions. These results thus shed some light on two potential issues. On the one hand, describing language similarities in terms of typological families is perhaps not sufficient for the purposes of this chapter. On the other hand, the multilingual

model architecture used in the tagging experiments of this chapter might not be sufficient to fully take advantage of language similarities.

7.4 Morphological Inflection

Having investigated two sequence labelling tasks, we now turn to a sequence-to-sequence prediction task, namely morphological inflection. The 2017 shared task on morphological inflection offers a large amount of data for 52 languages (Cotterell et al., 2017). Whereas the shared task has two sub-tasks, namely inflection and paradigm cell filling, we only evaluate on the inflection task. Furthermore, we use the high-resource setting, in which we have access to 10,000 training examples per language. The inflection subtask is to generate a target inflected form, given a lemma with its part-of-speech, as in the following example:

```
Source form and features: release V;NFIN
Target tag:               V;V.PTCP;PRS
Target form:              releasing
```

7.4.1 Method

We employ a deep neural network for the experiments in morphological inflection. This consists of an attentional sequence-to-sequence model, as described in Östling and Bjerva (2017).[9,10] The system takes embedded character representations as input to a Bi-LSTM encoder. The output of the encoder is passed through an attention mechanism, to an LSTM decoder which also takes the target form's morphological

[9] Available at `https://github.com/bjerva/sigmorphon2017`.

[10] In the SIGMORPHON shared task, this team placed as the 4th best (Cotterell et al., 2017).

tags as features. All layers in the network has 128 hidden units. Optimisation is done using Adam (Kingma and Ba, 2014) with default parameters. Whereas Östling and Bjerva (2017) explore learning a single model per language, in this chapter we experiment with learning joint models across languages. Additionally, we do not use an ensemble for the results presented in this chapter. The system architecture is visualised in Figure 7.2

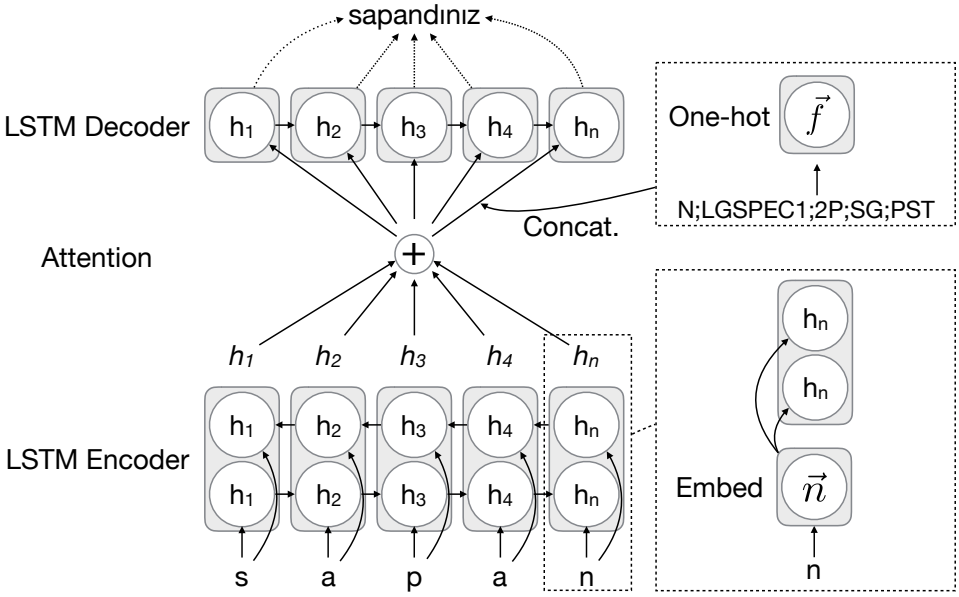


Figure 7.2: Architecture used for morphological inflection, consisting of an encoder-decoder with attention. The example depicts the production of the Turkish inflected form *sapandınız*, based on the input *sapan* and the tags *N;LGSPEC1;2P;SG;PST*.

Experimental Setup

We train our system using joint input and output representations. In order to examine the effect of adding a language to the mix, we train each model as follows. Given each language in the set of languages $l \in L$, we sample all language combinations l_1, l_2. We then train on the entire *high* dataset of l_1 (i.e., 10,000 examples), combined with

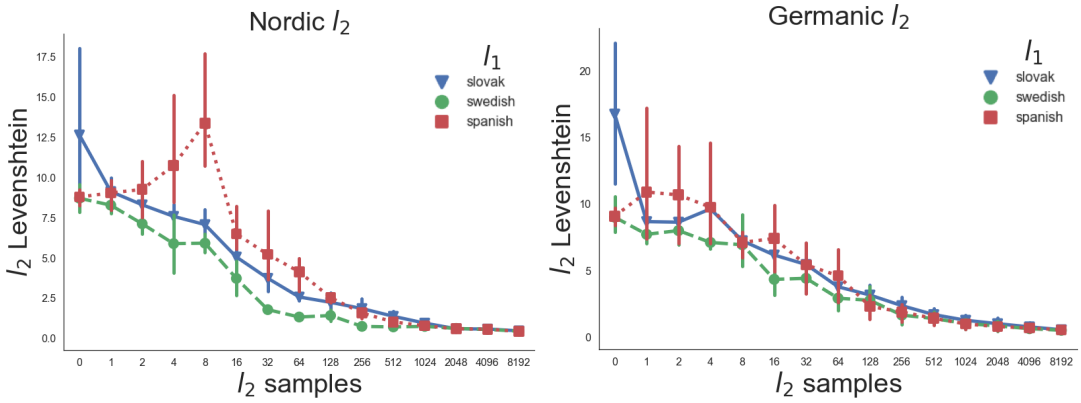
n training examples from l_2, with $n = [0, 2^0, 2^1, \ldots, 2^{13}]$.[11] In other words, l_1 is our source language, and l_2 our target language. This yields a total of $|L| \times |L| \times |n| = 24,000$ experiments. Note that the model is language agnostic, and apart from orthographic similarities between languages, has no way of knowing whether a certain string belongs to, e.g., Norwegian Nynorsk or Bokmål. We train each model for a total of 36 hours on a single CPU core, and report results using the model with the best validation loss.

7.4.2 Results and Analysis

Evaluation is done using the standard metric for this task, namely the Levenshtein distance between the predicted form and the target form (i.e. lower is better). Figure 7.3 shows results group-mean results on l_2 accuracy for training size n. The green lines show results when the $l_1 =$ Swedish, the red lines when the $l_1 =$ Spanish, and the blue when $l_1 =$ Slovak. Note that for performance is always better when training is combined with a language from the same language group. Notable is the performance with $l_1 =$ Spanish in the Nordic language group, where performance in fact drops when adding more l_2 samples at first. This indicates that transfer from languages which are more similar is beneficial, as compared to transfer from less related languages. This should come as no surprise, as the morphological similarities between, e.g., Norwegian and Danish are very pronounced, whereas similarities between Norwegian and Spanish are limited, if any exist. As a transfer baseline, the bottom right shows an average across all language families, showing that none of the languages are inherently better as source languages, as confidence intervals overlap for almost all amounts of l_2 samples.

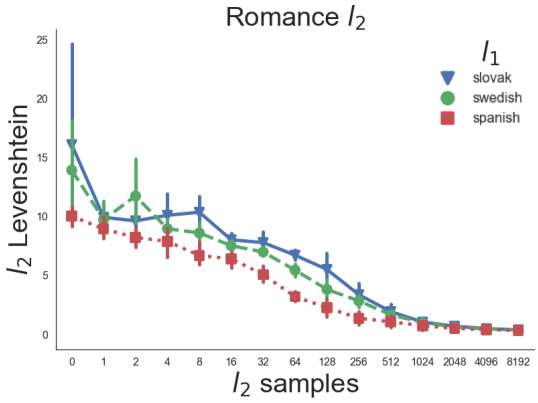
The results when using fewer than $256-512$ l_2 samples are, across the board, below baseline levels. This can be explained by the fact

[11] The SIGMORPHON-2017 shared task dataset contains three resource settings: low (100 examples), medium (1000 examples), and high (10000 examples).

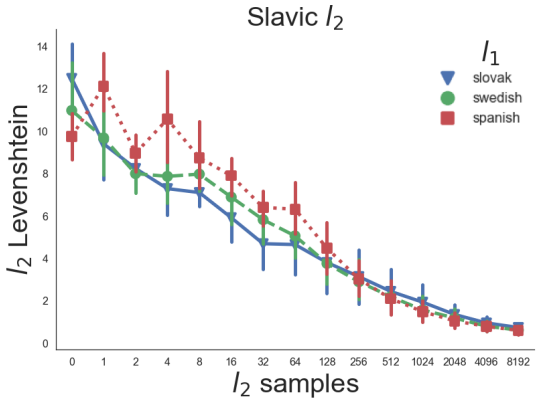


(a) Mean Nordic l_2 Levenshtein

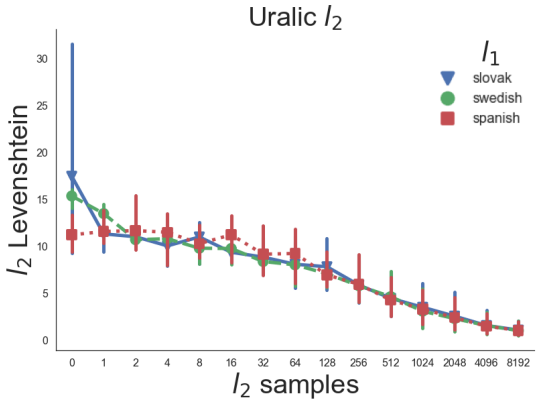
(b) Mean Germanic l_2 Levenshtein



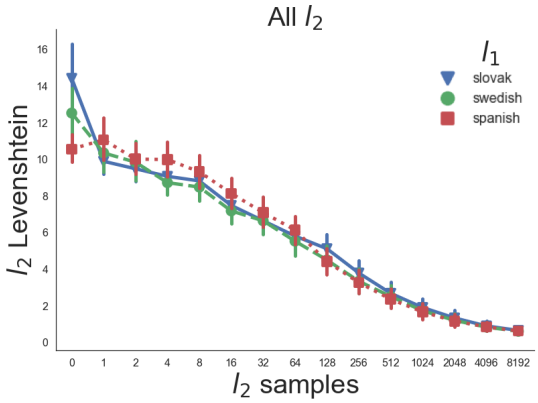
(c) Mean Romance l_2 Levenshtein



(d) Mean Slavic l_2 Levenshtein



(e) Mean Uralic l_2 Levenshtein



(f) Mean l_2 Levenshtein for all l_2s

Figure 7.3: Results on morphological inflection, with Slovak (blue, full, triangles), Swedish (green, dashed, circles), and Spanish (red, dotted, squares) as l_1.

that the system setup used in these experiments was not sufficient for cross-lingual transfer to be particularly successful. Changes in the architecture, such as including language vectors, as described by Östling and Tiedemann (2017) and Malaviya et al. (2017) is one possibility of improving this.[12]

In all cases, when sufficient l_2 data has been observed, the differences in performance with different l_1 is close to zero. This indicates that the model is not relying on information from the l_1 in such cases. The results obtained by the multilingual system when observing 2^{13} l_2 samples are similar to those obtained by the monolingual systems in Östling and Bjerva (2017). There are slight drops in performance across the board, which can be explained by two factors. On the one hand, some net capacity is wasted (from the perspective of l_2 performance), as we encode several languages in the same model. Additionally, we only observe 2^{13} l_2 samples, whereas the systems in Östling and Bjerva (2017) use all 10,000 samples available for training.

7.4.3 Summary of Results on Morphological Inflection

Similarly to the semantic tagging results, we observe that typologically related languages do tend to fare better in this transfer setting. Perhaps most convincing are the results when using Swedish as the source language and evaluating on Nordic target languages. This might be caused by the fact that the languages included in the Nordic (Danish, and Norwegian Bokmål and Nynorsk) group are highly similar to Swedish, whereas the other languages and language groups under consideration are more distinct.

[12] Language vectors are described further in Section 7.5.1.

7.5 Estimating Language Similarities

So far, we have considered the research questions dealing with the effects of hard parameter sharing between languages. The results have differed per task and per language combination, with the general trend that languages which *seem* similar, tend to be beneficial in combination with one another. This brings us to the final research question addressed in this chapter, namely, with what type of similarity measures does multilingual effectivity correlate?

As grouping by typological language families yielded a relatively large spread in results, one possibility is that language similarities should be quantified in a different manner. We investigate two different measures to estimate the similarity between languages. These measures have in common that they are very easily produced, meaning that they are not restricted to a few languages. In fact, the measures are readily available for a significant portion of the languages in the world. Furthermore, the two measures are quite different from one another – one directly obtained in a data-driven manner, and one based on edit distances on a lexical level.

7.5.1 Data-driven Similarity

The data-driven similarity measure which we employ is based on training language embeddings together with a Long Short-Term Memory language model (Hochreiter and Schmidhuber, 1997). The vectors are learned by conditioning the LSTM's prediction on an embedded language representation, when training the language model on a large collection of languages. This leads to the model learning representations which encapsulate some type of language similarity, which means that the vectors can be used to calculate similarities between languages, and is presented by Östling and Tiedemann (2017).[13] Furthermore, the method is applicable to languages with

[13]Thanks to Robert Östling for providing us with access to this resource.

very limited data, such as all languages with, for instance, a translation of the New Testament (i.e. ≈ 1000 languages). This approach to obtaining distributed can be compared to Malaviya et al. (2017), in which similar representations are learned in a neural machine translation system. We use the cosine distance between the vectors of two languages as a measure of their similarity.

7.5.2 Lexical Similarity

We calculate lexical similarity as in Rama and Borin (2015), by using normalised Levenshtein distance (LDN) between aligned word lists. LDN is calculated by summing length normalised Levenshtein distances for pairs of words using, e.g., Swadesh lists.[14] While effects such as similarity between phoneme inventories could cause unrelated languages to seem related, LDN has the advantage that it compensates for such effects (Rama and Borin, 2015).

Lexically aligned lists, similar to the Swadesh lists, are obtained from the Automated Similarity Judgement Program (ASJP) database (Wichmann et al., 2016).[15] The ASJP aims to offer 40-word lists for all of the world's languages, and currently offers such lists for 4664 languages.[16] These lists are linked on the meaning level, which allows for comparison of words across languages (see Table 7.9 for an example). The lists do not contain the orthographic representations of these words, but rather a phonemic representation. Such a representation is beneficial for our purposes, as differences in orthography resulting from historical artefacts might otherwise skew the results. For instance, while the orthographic representations of the 1st person singular pronoun in English and Norwegian have the

[14] Swadesh lists are standardised word lists, covering semantic concepts which are normally found in a given language, developed for the purposes of historical-comparative linguistics.

[15] `http://asjp.clld.org/`

[16] As of 11-05-2017.

Table 7.9: Examples from ASJP for English, Dutch, Norwegian, Finnish, and Estonian

Word/Meaning	English	Dutch	Norwegian	Finnish	Estonian
I	Ei	ik	yEi	minE	mina
you	yu	yEi	d3	sinE	sina
we	wi	vEi	vi	me	me
one	w3n	en	En	iksi	uks
two	tu	tve	tu	kaksi	kaks

maximum possible Levenshtein distance for the word pair (*I* vs. *jeg*), their phonemic representations reveal the commonalities (*Ei* vs *yEi*).

Figure 7.4 further illustrates the lexical distance measure. Languages which are typologically similar to each other are automatically grouped together, using the hierarchical clustering algorithm *UPGMA* (Unweighted Pair Grouping Method with Arithmetic-mean, cf. Saitou and Nei, 1987).

7.5.3 Results and Analysis

We will now consider the correlations observed between these language measures, and the results obtained from the multilingual experiments outlined in this chapter. The results from the semantic tagging are not included in this analysis, as we have too few data points available for the semantic tagging task to allow for reliably quantitative analysis. However, it is worth noting that the Germanic languages tend to help each other out, whereas Italian is generally less beneficial to performance.

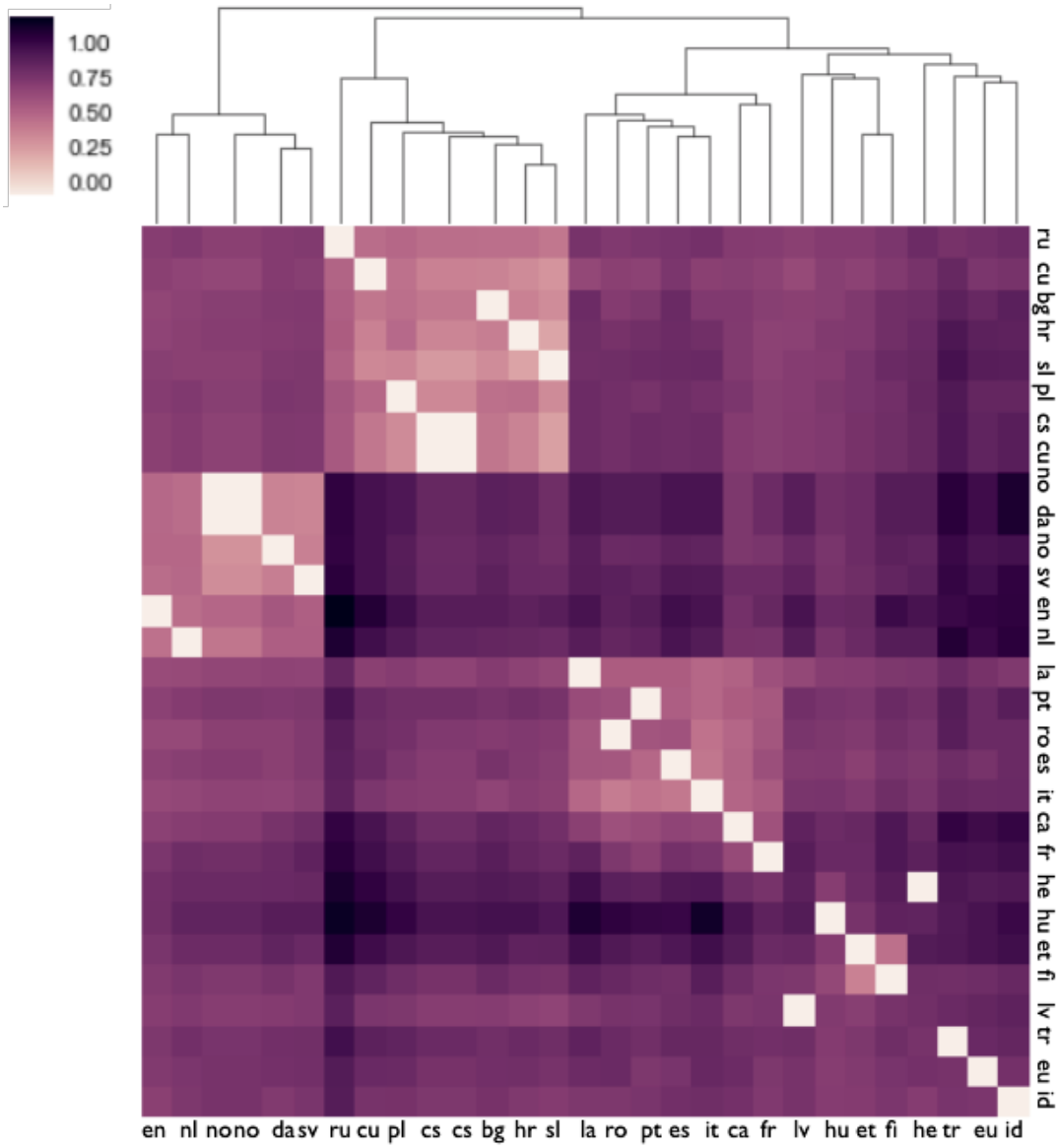


Figure 7.4: Distances calculated using LDN between ASJP lists, clustered with UPGMA.

Tagging Tasks in the Universal Dependencies

Figure 7.5a shows language correlations with language vector similarities, across languages and conditions in the PoS tagging task (Spearman $\rho = -0.14$ ($p = 0.001$)). Figure 7.5b contains the corresponding plot for the dependency relation task (Spearman $\rho = -0.19$ ($p \ll 0.001$)). Although these correlations are statistically significant, it is debatable whether or not they are practically significant. Given this amount of data points, statistically significant results are relatively likely, as the p-value indicates the risk of the correlation coefficient being equal to zero, given the data.

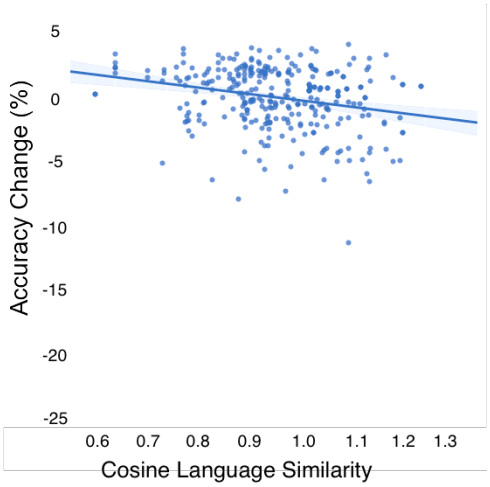
Although the correlations themselves are rather weak, it is interesting to observe that the patterns for both language similarities are rather similar. This is likely due to the fact that both of these measures offer some explanatory value for the problem at hand, and might also be a side-effect of the fact that these two measures correlate rather well with one another ($\rho = 0.7, p \ll 0.001$).

Morphological Inflection

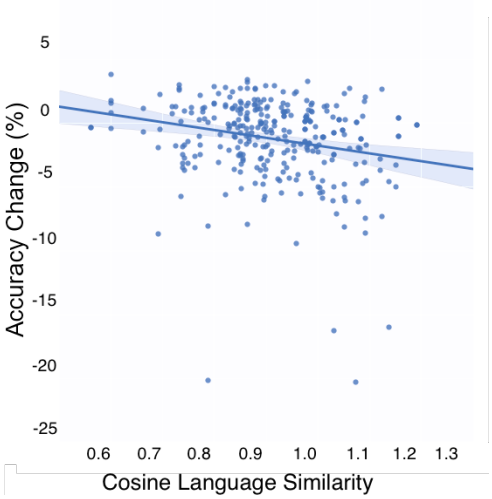
The correlations between vector distances and performance in morphological inflection are weak, as seen in Figure 7.6a (Spearman $\rho = 0.075$, $p < 0.001$). The correlation coefficient is somewhat higher when comparing with Levenshtein distance, as seen in Figure 7.6b (Spearman $\rho = 0.16$, $p \ll 0.001$).

7.5.4 When is Multilinguality Useful?

As we used relatively little data for training the tagging models, so as to allow for inclusion of a large number of languages, the absolute performance obtained is quite low. However, there appears to be some relation between the usefulness of adding in one more language to a model, and how similar those languages are. Although this seems is quite intuitive, the effects observed in our training set-

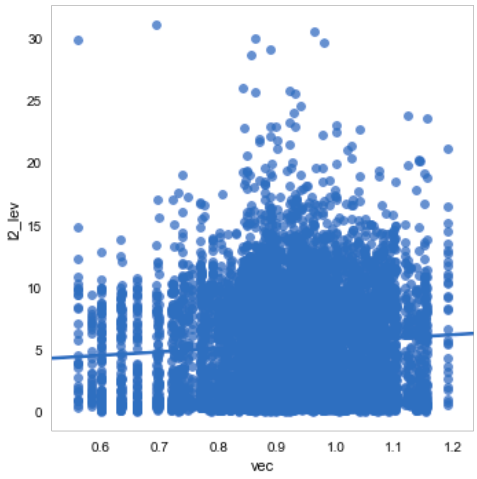


(a) Language vector distances compared to change in accuracy on PoS tagging.

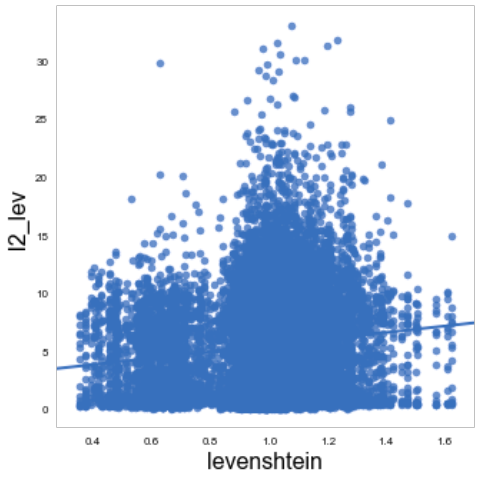


(b) Language vector distances compared to change in accuracy on dependency relation tagging.

Figure 7.5: Language vector distances: Correlations between accuracy and language similarities.



(a) Language vector distances.



(b) Levenshtein distances.

Figure 7.6: Morphological inflection: correlations between Levenshtein distance and language similarities.

ting were more subtle than expected. For instance, in many cases languages which are not particularly related appear to also increase system performance. This might be explained by two factors. On the one hand, it is possible that the quality of the word embeddings used is high enough so as to make the model fairly language agnostic. An alternative explanation, is that the network simply uses the information from a second language to further adjust its prior.

In the case of morphological inflection, we saw that the edit-distance based measure of language similarity was more informative than the language vectors. The fact that a measure based on edit distances is more successful here, is not altogether surprising as the task deals with minimising the Levenshtein distance between the predicted inflected form and the target form.

The effects seen in this work were weaker than expected, indicating that additional factors to language similarity as defined in this work govern the usefulness of multilinguality. However, the weak correlations still hold, indicating that choosing languages which are similar either in terms of lexical distance, or in terms of language vectors, might be a good place to start. An interesting prospect for future work, is to incorporate, e.g., the language vectors as a feature. This might make it easier for the model to learn between which languages it is the most beneficial to share certain parameters (e.g. between Nordic languages), and between which languages such sharing would likely lead to negative transfer.

7.6 Conclusions

We investigated multilinguality in four NLP tasks, and observed correlations between performance in multilingual models with two measures of language similarity, in addition to a preliminary comparison based on typological language families. On a general level, we found some cases in which using a source language related to

the target language was beneficial, mainly in the case of semantic tagging **RQ 4a**. We then looked at two measures of language similarities **RQ 4b**, which showed some correlations with multilingual model effectivity. The correlations found were, however, rather weak, indicating that language similarities as defined in this work are not sufficient for explaining such improvements to a large degree **RQ 4c**.

In the next chapter, we will nonetheless continue on this path, attempting to both exploit language similarities, as well as similarities between tasks **RQ 5**.