EHU at the SIGMORPHON 2016 Shared Task. A Simple Proposal: Grapheme-to-Phoneme for Inflection

Iñaki Alegria, Izaskun Etxeberria
IXA taldea, UPV-EHU
{i.alegria,izaskun.etxeberria}@ehu.es

Abstract

This paper presents a proposal for learning morphological inflections by a grapheme-to-phoneme learning model. No special processing is used for specific languages. The starting point has been our previous research on induction of phonology and morphology for normalization of historical texts. The results show that a very simple method can indeed improve upon some baselines, but does not reach the accuracies of the best systems in the task.

1 Introduction

In our previous work carried out in the context of normalization of historical texts (Etxeberria et al., 2016) we proposed an approach based on the induction of phonology. We obtained good results using only induced phonological weighted finite-state transducers (WFSTs), i.e. by leveraging the phoneme-to-grapheme method to yield a grapheme-to-grapheme model. The research question now is if the grapheme-to-grapheme model can be extended to handle morphological information instead of words or morphological segmentation. To assess this, we test a general solution that works without special processing for specific languages (i.e. we do not focus on special treatment of accents in Spanish and other idiosyncracies).

1.1 Task

We only have taken part in task 1 (Inflection from lemma/citation form) of the SIGMORPHON 2016 Shared Task (Cotterell et al., 2016). Given a lemma with its part-of-speech, the system must generate a target inflected form whose morphosyntactic description is given.[1]

1.2 Corpora and Resources

We use the data provided by the organizers of the task. Our first experiments and tuning were conducted on eight languages before the two additional 'surprise' languages (Maltese and Navajo) were provided.

We also ran experiments using the available bonus-resources (track 3) but after initial results we decided to present only a system using the basic resources.

2 Related work

In our previous work (Etxeberria et al., 2014; Etxeberria et al., 2016) we have used *Phonetisaurus*,[2] a WFST-driven phonology tool (Novak et al., 2012) which learns to map phonological changes using a noisy channel model. It is a solution that works well using a limited amount of training information. The task addressed earlier was the normalization of historical/dialectal texts.

In the same paper we demonstrated that the method is viable for language-independent normalization and we tested the same approach for normalization of Spanish and Slovene historical texts obtaining similar or better results than previous systems reported by Porta et al. (2013) (using hand-written rules) and Scherrer and Erjavec (2015) (using a character-based SMT system).

Because of the model's relative success with historical normalization and its simplicity, we developed the approach further for addressing the shared task problem.

There exist other finite-state transducer-based approaches, generally more complex than what we present, of which two warrant a mention:

(i) Dreyer et al. (2008) develops a model for

[1] http://www.sigmorphon.org/sharedtask

[2] https://github.com/AdolfVonKleist/Phonetisaurus

string-to-string transduction where results are improved using latent-variables.

(ii) Cotterell et al. (2015) models word-forms using latent underlying morphs and phonology. The system includes finite-state technology (in the form of WFSA and PFSTs) in two of the three steps: concatenation, phonology, and phonetics.

3 Experiments and Evaluation

3.1 Basic Method

We used *Phonetisaurus* to train a WFST-system that learns the changes that occur when going from the citation form to another form. This tool—while not specifically limited to such uses—is widely used for rapid development of high-quality grapheme-to-phoneme (g2p) converters. It is open-source, easy-to-use, and authors report promising results (Novak et al., 2012).

Phonetisaurus uses joint n-gram models and it is based on OpenFST, which learns a mapping of phonological changes using a noisy channel model. The application of the tool includes three major steps:

1. Sequence alignment. The alignment algorithm is based on the algorithm proposed in Jiampojamarn et al. (2007) and includes some minor modifications to it.

2. Model training. An n-gram language model is trained using the aligned data and then converted into a WFST. For producing the language model, we used the Language Model training toolkit *NGramLibrary* for our experiments, although several alternative similar tools exist that all cooperate with *Phonetisaurus*: *mitlm, NGramLibrary,* SRILM, SRILM *MaxEnt extension, CMU-Cambridge SLM*.

3. Decoding. The default decoder used in the WFST-based approach finds the best hypothesis for the input words given the WFST obtained in the previous step. It is also possible to extract a k-best list of output hypotheses for each word.

The alignment algorithm is capable of learning many-to-many relationships and includes three modifications to the basic toolkit: (a) a constraint is imposed such that only many-to-one and one-to-many alignments are considered during training; (b) during initialization, a joint alignment lattice is constructed for each input entry, and any unconnected arcs are deleted;[3] (c) all transitions, including those that model deletions and insertions, are initialized with and constrained to maintaining a non-zero weight.

As the results obtained with this tool were the best ones in our previous scenario, we decided to employ it for this task. Concretely, we have used *Phonetisaurus* to learn a WFST which can translate simplified morphological expressions to words to solve the inflection task. Once the transducer is trained, it can be used to generate correspondences for previously unseen morphological representations and their corresponding word-forms.

3.2 Testing the models

Using the development section for tuning we experimented with different variations in our approach in order to tune a good model for the problem.

First, we compacted the morphological information in a tag (which we consider a pseudo-morpheme) by concatenating the first letter in the category with a consecutive number. For example, the first lines in the training corpus for German

aalen pos=V, ... per=1,num=PL	aalen
aalen pos=V, ... per=3,num=PL	aalen
aalen pos=V, ... per=2,num=SG	aaltest
aalen pos=V, ... per=3,num=SG	aalte
aalen pos=V,tense=PRS	aalend

are converted into:

aalen V0	aalen
aalen V1	aalen
aalen V2	aaltest
aalen V3	aalte
aalen V4	aalend

Using this information three experiments were carried out where the morphosyntactic information was

- treated as a suffix.
- treated as a suffix and as a prefix.
- treated as a suffix, as an infix in the center of the lemma, and as a prefix.

[3] The topology of the WFST is assigned by the tool and the model is rather large (standard parameters are used: from 1-gram to 7-gram).

The strongest results were obtained using the second model for all languages except Finnish, which yielded the best results using only a suffix-based representation.

To illustrate the encoding, below are the first few entries in the development corpus for German:

N96+Aak+N96	→	*Aak*
V87+aalen+V87	→	*geaalt*
V79+aasen+V79	→	*aaste*
V1+abandonnieren+V1	→	*abandonnieren*
A40+abchasisch+A40	→	*abchasischerem*

In a second step we built different WFSTs depending on the category, but this yielded no improvement. As an alternative, we decided to test if putting only the category information in the prefix (i.e. one character) could help in the task. This produced an improvement only for Finnish.

As a third step we tested the possibility of optimizing the size and the content of the tag (the pseudo-morpheme), attempting to match its length with the length of the corresponding morpheme, as in the following example for German encodings:

N+Aak+N96	→	*Aak*
V87+aalen+V87	→	*geaalt*
V+aasen+V79	→	*aaste*
V+abandonnieren+V1	→	*abandonnieren*
A+abchasisch+A4000	→	*abchasischerem*

This strategy produced no solid improvement in our preliminary experiments.

3.3 Evaluation

We have measured the quality using the metrics and the script provided by the organizers; the baseline figures also originate with the organizers.

In all the languages whole tags were injected as prefixes and suffixes, with the exception of Finnish, where in the prefix tag position only the first character is included. For example, for the wordform *aakkostot* ‘alphabets’ *N+aakkosto+N9* is used instead of *N9+aakkosto+N9*.

For the submitted final test we retrained the transducer adding the development section to the training corpus. As can be seen in table 1, a slight improvement was obtained (0.43% on average).

4 Using external information

Trying to take advantage of bonus resources, we used a word list for Spanish, German and Russian available with the *FreeLing* package (Carreras et al., 2004) as a 1-gram language-model of words.

Language	Baseline	Dev	Test
Arabic	**69.40**	67.53	64.68
Finnish	69.80	**86.86**	83.72
Georgian	**91.60**	87.04	83.11
German	89.90	**91.61**	89.86
Hungarian	74.10	**91.04**	85.39
Maltese	36.56	**61.89**	64.80
Navajo	70.30	**93.53**	56.33
Russian	**90.20**	86.74	86.58
Spanish	**95.49**	90.98	91.35
Turkish	59.20	**90.36**	90.84
Mean		**84.76**	79.67

Table 1: Results on the test corpus using 1-best accuracy for evaluation.

Since it is possible to produce multiple outputs from the WFST we train, we also experimented with an approach where the WFST would return several ranked candidates (3, 5, and 10), and selecting the first one found in the word list. If none of the candidates appeared in the list, the first proposal was used.

Using this strategy the results for Spanish improved slightly (by 2%), while the results for German improved slightly less (by 0.2%), and the Russian results worsened (by -0.7%).

Language	Basic	Filtering 3	Filtering 5
German	91.61	**91.80**	91.73
Russian	**86.74**	86.05	84.73
Spanish	90.98	**92.86**	92.86

Table 2: Accuracy when using a word list for filtering the proposals from the WFST. The first column shows the results without any external resources used; in the second column a word list has been used for filtering the top 3 proposals and in the third column for filtering with the top 5 proposals.

Since *FreeLing* is known to produce the highest-quality output for Spanish, we may assume that the results reflect the relative quality of the resources in that package.

Due to this limited improvement, we decided to present only the basic system for track 1.

5 Conclusions and future work

Previous work on lexical normalization on historical and dialectal texts has been extended and applied to a morphological inflection scenario.

While the method is simple and somewhat limited, with results not fully competitive against the best reported systems (Cotterell et al., 2016), some difficult languages saw a relatively good performance (Navajo and Maltese).

In the near future, our aim is to improve the results by trying to place the tags and morphemes in a more congenial configuration for WFST training and to use existing proposals to harness available latent information (Dreyer et al., 2008). In addition to this, we plan to incorporate techniques learned from other participants in the shared task.

References

Xavier Carreras, Isaac Chao, Lluis Padró, and Muntsa Padró. 2004. FreeLing: An open-source suite of language analyzers. In *Proceedings of LREC*.

Ryan Cotterell, Nanyun Peng, and Jason Eisner. 2015. Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3:433–447.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany, August. Association for Computational Linguistics.

Markus Dreyer, Jason R Smith, and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1080–1089. Association for Computational Linguistics.

Izaskun Etxeberria, Inaki Alegria, Mans Hulden, and Larraitz Uria. 2014. Learning to map variation-standard forms using a limited parallel corpus and the standard morphology. *Procesamiento del Lenguaje Natural*, 52:13–20.

Izaskun Etxeberria, Inaki Alegria, Larraitz Uria, and Mans Hulden. 2016. Evaluating the noisy channel model for the normalization of historical texts: Basque, Spanish and Slovene. In *Proceedings of the LREC2016*.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *HLT-NAACL*, volume 7, pages 372–379.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. 2012. WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49, Donostia–San Sebastian, July. Association for Computational Linguistics.

Jordi Porta, José-Luis Sancho, and Javier Gómez. 2013. Edit transducers for spelling variation in old Spanish. In *Proc. of the workshop on computational historical linguistics at NODALIDA 2013. NEALT Proc. Series*, volume 18, pages 70–79.

Yves Scherrer and Tomaž Erjavec. 2015. Modernising historical Slovene words. *Natural Language Engineering*, pages 1–25.