Predicting Pronouns with a Convolutional Network and an N-gram Model

Christian Hardmeier
Uppsala University
Department of Linguistics and Philology
751 26 Uppsala, Sweden

Abstract

This paper describes the UU-HARDMEIER system submitted to the DiscoMT 2017 shared task on cross-lingual pronoun prediction. The system is an ensemble of convolutional neural networks combined with a source-aware *n*-gram language model.

1 Overview

For the 2017 cross-lingual pronoun prediction shared task, we chose to create a system that could be implemented very quickly while still providing an interesting comparison to the other systems we expect to participate in the shared task. The core components of our system are a convolutional neural network that evaluates the context of the source and target context of the examples. As in our systems from the previous year (Hardmeier, 2016; Loáiciga et al., 2016), we also use a source-aware *n*-gram language model as a complementary component. In contrast to 2016, our neural network classifier does not attempt to model pronominal anaphora explicitly. This change was made to simplify the model and avoid the heavyweight preprocessing that our earlier systems required. Instead, we focused on implementing a more sophisticated system combination method that permits the construction of a larger ensemble of models.

2 Convolutional neural network

The neural network architecture of our pronoun prediction model is loosely inspired by the winning system of the WMT 2016 shared task on cross-lingual pronoun prediction (Luotolahti et al., 2016). However, since we expected a large proportion of the participating systems to use recurrent neural networks, we decided to use a simpler convolutional architecture instead. The implementation of the network uses the Keras library (Chollet et al., 2015).

The network independently scans four different input areas for each example: *left source*, *left target*, *right source* and *right target*. All four areas are defined with respect to the position of the element to be predicted, which is a placeholder to be filled on the *target* side aligned to a pronoun on the *source* side. The *left* areas cover the context preceding the pronoun or placeholder, up to the beginning of the previous sentence or at most 50 tokens to the left of the anchoring position, whichever is shorter. The *right* areas cover the context following the pronoun or placeholder, up to the end of the current sentence or at most 50 tokens if the sentence is longer. The context size limit of 50 tokens is large enough to have no effect in most cases, but it ensures that the training efficiency does not suffer from a few overlong sentences. The source language pronoun aligned to the placeholder is included in both the *left* and *right source* context area, whereas the placeholder on the *target* side is excluded from the context areas.

The words of the source and target language are encoded as one-hot vectors using the vocabulary of the IWSLT part of the official training data. Words occurring only once in the IWSLT training set are excluded from the vocabulary and treated as unknown words instead. The part-of-speech tags provided in the training data are ignored. The one-hot vectors are mapped to dense embeddings through an embedding layer with *tanh* activation, whose weights are initialised randomly at training time.

The dense word embeddings form the input of one convolutional layer per input area. The output of the convolutional layers undergo max pooling in a single step over the entire length of the input area. Then the vectors resulting for the four input areas are concatenated together and used as the input of a densely connected layer with softmax activation

Proceedings of the Third Workshop on Discourse in Machine Translation, pages 58–62,
Copenhagen, Denmark, September 8, 2017.

	Network properties					*Epochs included*			
	Training	Weighting	Optimiser	Minibatch	Conv. filters	de-en	en-de	en-fr	es-en
A	all	–	Adam	100	100	1	3	3	–
B	IWSLT	+	Adam	100	100	15	20	20	20
C	IWSLT	–	Adam	100	100	–	–	–	20
D	IWSLT	+	rmsprop	20	50	–	–	–	20
E	IWSLT	+	Adam	20	50	–	–	–	20

Table 1: Properties of the convolutional neural networks included in our submissions

that predicts the class of the example.

We trained the convolutional neural network in different configurations. Five configurations were included in some form in our submissions to the shared task. Unfortunately, we worked under very strong time pressure, and the selection of the included systems and the exploration of the parameter space is not as systematic as we should have wished. We here describe the systems as submitted, without making any specific claims regarding the usefulness of the parameter settings we tested. Also, we did not have time to train the selected systems to convergence. Instead, we saved a snapshot of the network weights after each completed training epoch and ran all these snapshots on the test data. Then we left it to the system combination procedure described in Section 4 to assign weights to all the different snapshots according to their usefulness measured on the development set.

Table 1 shows an overview of the properties distinguishing the five systems used in the submissions and the number of epochs per system included for each language pair (limited by the available training time). Parameters common to all systems are not listed in the table. These include the word embedding in the source and target languages, which were set to 100, and the kernel size in the convolutional layer, which was set to 10.

System A was trained on all training data provided by the organisers, but could only complete a small number of training iterations. The other systems are trained on IWSLT data only. Systems B, D and E use an example weighting scheme that attempts to assign equal weight to all classes in the data regardless of their frequency. Systems A, B, C and E were trained with the Adam optimiser using the default settings in Keras (learning rate 0.001), whilst system D was trained with rmsprop and a learning rate of 0.01. The minibatch sizes were 100 and 20 for different systems, and the number of convolutional filters were 100 and 50.

3 Source-aware language model

In our submissions to the WMT 2016 shared task on cross-lingual pronoun prediction (Hardmeier, 2016; Loáiciga et al., 2016), we found that a simple n-gram language model extended with access to the identity of the source pronoun achieved quite good results in comparison to our more sophisticated neural network classifier. The information captured by this model seemed to be complementary to that encoded in the neural network, so that additional gains could be realised by combining the two models. This year, we again use a source-aware language model as a component in our work. The following description follows our earlier system description paper (Hardmeier, 2016) and is repeated here for reference.

Our source-aware language model is an n-gram model trained on an artificial corpus generated from the target lemmas of the parallel training (Figure 1). Before every REPLACE tag occurring in the data, we insert the source pronoun aligned to the tag (without lowercasing or any other processing). The alignment information attached to the REPLACE tag in the shared task data files is stripped off. In the training data, we instead add the pronoun class to be predicted. The n-gram model used for this component is a 6-gram model with modified Kneser-Ney smoothing (Chen and Goodman, 1998) trained with the KenLM toolkit (Heafield, 2011) on the complete set of training data provided for the shared task.

To predict classes for an unseen test set, we first convert it to a format matching that of the training data, but with a uniform, unannotated REPLACE tag used for all classes. We then recover the tag annotated with the correct solution using the `disambig` tool of the SRILM language modelling toolkit (Stolcke et al., 2011). This tool runs the Viterbi algo-

Source:	**It** 's got these fishing lures on the bottom .
Target lemmas:	**REPLACE_0** avoir ce leurre de pêche au-dessous .
Solution:	*ils*
LM training data:	It **REPLACE_*ils*** avoir ce leurre de pêche au-dessous .
LM test data:	It **REPLACE** avoir ce leurre de pêche au-dessous .

Figure 1: Data for the source-aware language model

rithm to select the most probable mapping of each token from among a set of possible alternatives. The map used for this task trivially maps all tokens to themselves with the exception of the REPLACE tags, which are mapped to the set of annotated REPLACE tags found in the training data.

In addition to being included as a component in our primary ensemble systems, we submitted the output of the standalone source-aware language model as a secondary submission for all languages.

4 System combination

To combine the neural predictor with the source-aware language model, we linearly interpolated the probabilities assigned to each class by each model. The class finally predicted was the one that scored highest according to the interpolated probability distribution.

The neural network prediction probabilities are obtained trivially as the posterior distribution of the final softmax layer of the convolutional network. For the source-aware language model, we run SRILM's `disambig` tool with the `-posteriors` option, which causes it to output an approximate posterior distribution derived from information collected during the Viterbi decoding pass. For all classes c, the probability predicted by the combined model is defined as a convex combination of the probabilities $p_i(c)$ predicted by each model individually:

$$p(c;\lambda) = \sum_i \lambda_i \, p_i(c) \qquad (1)$$

To estimate the parameter vector λ, we maximise the log-likelihood of the interpolated model on a development set. The log-likelihood is defined as follows:

$$L(\lambda) = \sum_i \sum_c t_{ic} \log p(c;\lambda) \qquad (2)$$

Here, the index i ranges over the examples in the development set and c ranges over the classes. The indicator variable t_{ic} equals 1 if class c is the correct prediction for example i and 0 otherwise.

The parameter vector λ is then obtained as the solution of the following constrained optimisation problem:

$$\begin{aligned} &\text{Maximise } L(\lambda) \\ &\text{subject to } \textstyle\sum_k \lambda_k = 1 \text{ and } \lambda_k \geq 0 \text{ for all } k. \end{aligned}$$

To solve this problem, we apply the sequential least squares programming (SLSQP) algorithm (Kraft, 1988) as implemented in the SciPy library[1]. The resulting weights are then rounded to 4 decimals and component systems whose weight after rounding equals zero are discarded.

5 Results

The results of the official evaluation are shown in Table 2. In this paper, we concentrate on discussing our own systems. For an overview of the shared task results, see the report by Loáiciga et al. (2017). We note that the ensemble system improves over the source-aware n-gram model for all language pairs. The gap in macro-averaged recall exceeds 10 percentage points for German–English and Spanish–English. For English–French, it is about 4 points, and for English–German about 1.5. The results in terms of accuracy show a similar pattern. In Table 3, we find the weights assigned to the individual systems by the system combination procedure. Recall that the ensemble contains multiple instantiations of each of these models (see Table 1); here, the weights are summed over all epochs of a particular model. We observe that the interpolation method assigns appreciable weights to both the neural and the n-gram components in all languages, so that both models make a contribution to the final prediction. The English–German system has the highest language model weight, which partly explains the similar performance of the primary and the contrastive system for this language pair.

[1] http://www.scipy.org/

	Macro-R		Accuracy	
	prim.	contr.	prim.	contr.
de-en	62.18	51.12	79.49	69.23
en-de	58.41	56.80	71.20	69.02
en-fr	62.86	58.95	73.48	71.82
es-en	52.32	42.19	54.10	46.45

Table 2: Official evaluation results for primary (ensemble) and contrastive (n-gram) systems

	de-en	en-de	en-fr	es-en
LM	0.5437	0.7676	0.4552	0.2931
A	0.1886	0.0062	0.2957	–
B	0.2677	0.2262	0.2490	0.1825
C	–	–	–	0.3339
D	–	–	–	0.0674
E	–	–	–	0.1230

Table 3: Weights summed over all epochs for individual systems

Figure 2 shows the development performance of the individual neural networks included in the ensemble for German–English. The size of the dots on the accuracy curve is proportional to the interpolation weight. The figure suggests that system B, which is trained on IWSLT data only, is overfitting the training set and probably needs more regularisation. On the other hand, the performance of system A, trained on the full data set, still improves after 3 epochs, and it is likely that we could have achieved better results with more time for training.

A look at the confusion matrices for the different language pairs (not shown for space reasons) suggests that the convolutional neural networks manage to capture some relevant linguistic information from the context that the n-gram model misses. In particular, the ensemble systems for German–English and English–French are much more successful at distinguishing pronoun classes that require knowledge of the antecedent. In previous work (Hardmeier et al., 2013; Hardmeier, 2014), we used performance on the French pronouns *ils* and *elles* as an indicator of a system's capacity to reason about antecedents. Both pronouns are straightforward translations of the English pronoun *they*, differing in gender only. Our English–French ensemble achieves class F-scores of 76.19% (*elles*) and 83.54% (*ils*) on these classes, as opposed to

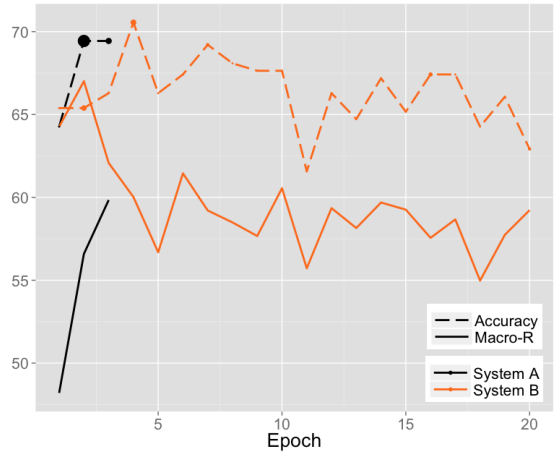

Figure 2: Development set performance of individual snapshots for German–English

35.29% and 79.01% for the n-gram system; this is a large improvement especially for *elles*. The German–English system faces similar difficulties for the pronouns *they* vs. *she* (Hardmeier and Federico, 2010) and likewise improves from 24.00% (*she*) and 56.18% (*they*) to 66.67% and 76.60%. In the other two language pairs, we find no such clear patterns. The predictions for English–German are almost the same in both systems, and Spanish–English improves much more uniformly over all classes.

6 Conclusions

The system described in this paper was created to provide an additional point of comparison in the shared task evaluation. It uses a very simple convolutional neural network architecture that can be contrasted with the more sophisticated neural models seen in the previous edition of the shared task. The source-aware *n*-gram model is another approach that achieved reasonable results in the previous evaluation. In comparison with last year, we now apply a better system combination procedure that permits the integration of a large number of systems in the final ensemble.

Acknowledgements

This work was supported by the Swedish Research Council under grant 2012-916 *Discourse-Oriented Statistical Machine Translation*. Computational resources were provided by CSC – IT Center for Science, Finland, through the Nordic Language Processing Laboratory (NLPL).

References

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Computer Science Group, Harvard University, Cambridge (Mass.).

François Chollet et al. 2015. Keras. `https://github.com/fchollet/keras`.

Christian Hardmeier. 2014. *Discourse in Statistical Machine Translation*, volume 15 of *Studia Linguistica Upsaliensia*. Acta Universitatis Upsaliensis, Uppsala.

Christian Hardmeier. 2016. Pronoun prediction with latent anaphora resolution. In *Proceedings of the First Conference on Machine Translation (WMT16)*. Association for Computational Linguistics, Berlin (Germany).

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *Proceedings of the Seventh International Workshop on Spoken Language Translation (IWSLT)*. Paris (France), pages 283–289.

Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2013. Latent anaphora resolution for cross-lingual pronoun prediction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle (Washington, USA), pages 380–391. http://www.aclweb.org/anthology/D13-1037.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh (Scotland, UK), pages 187–197. http://www.aclweb.org/anthology/W11-2123.

Dieter Kraft. 1988. A software package for sequential quadratic programming. Technical report, Institut für Dynamik der Flugsysteme, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt.

Sharid Loáiciga, Liane Guillou, and Christian Hardmeier. 2016. It-disambiguation and source-aware language models for cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation (WMT16)*. Association for Computational Linguistics, Berlin (Germany).

Sharid Loáiciga, Sara Stymne, Preslav Nakov, Christian Hardmeier, Jörg Tiedemann, Mauro Cettolo, and Yannick Versley. 2017. Findings of the 2017 DiscoMT shared task on cross-lingual pronoun prediction. In *Proceedings of the 3rd Workshop on Discourse in Machine Translation*.

Juhani Luotolahti, Jenna Kanerva, and Filip Ginter. 2016. Cross-lingual pronoun prediction with deep recurrent neural networks. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Association for Computational Linguistics, pages 596–601. https://doi.org/10.18653/v1/W16-2353.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*. Waikoloa (Hawaii, USA).