RACAI Entry for the IWSLT 2016 Shared Task

Sonia Pipa, Alin Florentin Vasile, Ioana Ionaşcu, Stefan Daniel Dumitrescu, Tiberiu Boros

Research Institute for Artificial Intelligence "Mihai Drăgănescu"
Romanian Academy, Bucharest
sonia@racai.ro, alin@racai.ro, ioana@racai.ro, sdumitrescu@racai.ro, tibi@racai.ro

Abstract

Spoken Language Translation is currently a hot topic in the research community. This task is very complex, involving automatic speech recognition, text-normalization and machine translation. We present our speech translation system, which was compared against the other systems participating in the IWSLT 2016 Shared Task. We introduce our ASR system for English and our MT system for English to French (En-Fr) and English to German (En-De) language pairs. Additionally, for the English to French Challenge we introduce a methodology that enables the enhancement of statistical phrase-based translation with translation equivalents deduced from monolingual corpora using neural word embedding.

1. Introduction

This paper describes the RACAI entry for the IWSLT 2016 Shared Task. It consisted of three tracks: automatic speech recognition, machine translation and speech translation. The data used in the challenge came from two different domains: recorded TED talks and recorded Skype conversations[1]. While the recorded TED talks fall within the narrative domain, recorded Skype calls belong to the oral domain. They contain a lot of hesitations which makes it hard for speech recognition systems to successfully make use of a language model for generating correct transcriptions. Our presentation will cover an automatic speech recognition (ASR) system designed for the English TED Talks transcription track, a text-normalization technique based on a hybrid neural network/n-gram model approach and a machine translation (MT) system that convers the En-Fr and En-De language pairs. For the En-Fr language pair we took the opportunity to test a new methodology that allowed us to automatically infer and introduce new translation equivalents using data extracted from a small translation dictionary and monolingual corpora.

Although we previously participated in the IWSLT Shared Task, in our previous attempts we only took part in the Machine Translation (MT) track. This is the first time we attempted other tracks in the challenge. Because of this, many of our tools and resources have not been previously evaluated on this type of data and the overall complexity of the evaluation campaign forced us to resort to baseline systems in our approach. However, this participation was important for us and provided a good contrastive evaluation metric for our future developments and participation in similar events.

2. Speech transcription

For the IWSLT 2016 shared task on ASR we focused on automatically transcribing pre-recorded TED Talks. The participants were given the opportunity to experiment with any available speech resources except for a number of TED Talks that were provided as a list by the organizers.

Our speech recognition software is a standard HMM-based on the Sphinx decoder (Lamere et al., 2003). The acoustic feature vector is composed of Mel-Frequency Cepstral Coefficients (MFCC) and their delta and delta-delta coefficients. For the acoustic model we relied on the VoxForge English corpus which is composed of 129 hours and 30 minutes of pre-recorded speech prompts from 1213 speakers.

3. Domain adapted language model

The overall performance of the ASR system depends on both the robustness of the acoustic model and on the quality of the language model (LM) it uses. Thus, carefully crafting the corpus on which one builds a LM for continuous speech recognition is of high importance for the task of speech transcription. It is important that the training corpus is of the same domain on which the speech transcription system is used.

Both quality and quantity weigh heavily on the performance of the computed n-gram probabilities. In order to build our corpus we used a bootstrapping method that allowed us to incrementally grow our available text resources. The method is perplexity-based and we previously used the same methodology for enhancing machine translation LM corpora. The procedure is the following:

(a) Given tokenized and true-cased English text from the unrestricted TED Talks, we built a 5-gram, Knesser-Ney smoothed LM.
(b) Next we merged all our available monolingual corpora into a single file. The monolingual corpora consisted of the given QED (Ahn et al., 2005), MultiUN (Eisele and Chen, 2010), Wikipedia, DGT (Steinberger et al., 2013) and some random corpora crawled from multiple websites (mostly news)). In total, we had 168 million sentences with 28 billion tokens.
(c) We then computed the sentence-level perplexity against the in-domain LM (a) for every sentence inside the new corpus (b), finally sorting the sentences (lower perplexity is better).
(d) We observed that after 10 million sentences the perplexity score rose and the sentences were mostly from the administrative domain. We kept only the first 10 million sentences and we concatenated them with the original TED corpus. We then computed a trigram LM which we further used in our transcription system.

3.1. ASR output text normalization

An un-normalized text is not directly usable in natural language processing applications, because it needs to be sentence-split, tokenized, word cased, etc., and then annotated

[1] https://sites.google.com/site/iwsltevaluation2016/mt-track

(part-of-speech tagged, chunked and parsed). Text normalization is extremely important for automatic machine translation (MT), speech-to-speech translation, information extraction, dialog systems, etc.

The importance of text normalization has yielded a large number of studies and research papers. Most of the methods rely on language modeling with n-gram models, but the particular details of implementations vary. As such, Israel et al. (2012) use an n-gram model built on words and POS tags and obtain an accuracy of 61.4%. Wang et al. (2013) interpolate 3-gram probabilities in order to analyze a window of 5 words, but they apply their method not for ASR output but on social media text normalization, on which they obtain an accuracy of 77.8%. Similar methods are also employed for Tweeter text normalization (82.24%) (Sonmez and Ozgur, 2014) and SMS text normalization (80.70%) (Aw et al., 2006).

Some authors also employ hybrid approaches based on language-specific rule-based and statistical phrase-based post-editing (Schlippe et al., 2010).

Our method for text normalization is a hybrid approach using an n-gram model for truecasing and a deep neural network (DNN) classifier trained with unsupervised word embeddings for punctuation restoration.

Truecasing has been previously done using n-gram models and this methodology is known to provide stable results. Furthermore, when using a wide-coverage training corpus one can make use of heuristics like the fact that unknown words are likely to be proper names or uncommon abbreviations and acronyms which must be either capitalized or uppercased. However, Large Vocabulary Speech Recognition (LVSR) is usually limited by its dictionary and out-of-vocabulary (OOV) words that end up being mapped to similar sounding groups of words. That is why we limited our approach to only relying on n-gram and we did not use any suffix or prefix analysis of OOV words which could theoretically yield higher accuracies. However, we intend to investigate this approach in a future work.

On the other hand, punctuation restoration has known only limited success when n-grams are applied. One observation is that punctuation marks, along functional words, are very frequent in any language, thus, when applying any type of smoothing over the n-gram probabilities, high frequency unigrams such as comma or period tend to radically increase the probability of n-grams which contain them and disable the possibility of accurately using comparisons between probabilities of sequences with and without punctuation. In fact, one of our early experiments concluded that if we interpolate 3-gram probabilities over a window of 5 tokens and try to estimate comma insertion probabilities based on this score we only get an F-score of 0.56, because the system tends to add as many commas as possible.

Given the above mentions, our text normalization methodology has two steps: first we establish correct word-casing using an n-gram model, and then we use a DNN classifier to determine punctuation insertion points within the text.

Given a sentence, our truecaser works by sequentially processing each word to determine its correct orthographic form.

The analysis process uses a window of 5 tokens centered on the word being processed. For word w_k we take into consideration words w_{k-2} to w_{k+2}. We try alternate orthographic forms for the words inside the feature window. Because words w_{k-2} and w_{k-1} have previously been processed we build the Cartesian product of spellings for the words w_k, w_{k+1} and w_{k+2}. The spellings refer to the 3 cases: lowercase, capitalized and uppercase. Thus, our system tests 27 possible combinations. For every combination we interpolate the probability of seeing that particular 5-word window using an n-gram model, as a dot product over a sliding window of size 3. This means that we calculate the group probability as a dot product between 3 n-gram probabilities: $P(w_{k-2}, w_{k-1}, w_k)$, $P(w_{k-1}, w_k, w_{k+1})$, $P(w_k, w_{k+1}, w_{k+2})$. The probabilities are computed from the training corpus and probability smoothing is applied to better handle unseen n-grams.

To build our n-gram model we used a Wikipedia English corpus composed 125.138.883 sentences, 3.035.591.789 words, 111.247.856 dots and 70.199.700 commas. The corpus was tokenized and we computed unigram, bigram and trigram counts. To test the functionality of our system we kept aside a random test set of 100 sentences. This subset was stripped of punctuation marks and all words were converted into their lowercase form. This enabled us to evaluate the performance of our system by seeing if it is capable of restoring the text to its original form. Accuracy does not correctly reflect the ability of the system to perform truecasing, thus, we measured both the success rate of the words that were changed to a different orthographic form, as well as the number of tokens that were correctly changed versus the number of tokens that should have been changed, but were left untouched by the system. Table 1 shows the detailed results on the test set.

Table 1 – Truecaser performance on the test set

Words	Precision	Recall	F-score
Capitalized / Uppercase word	0.79	0.83	0.81

Punctuation restoration requires a different approach than that of truecasing. As previously mentioned n-gram models do not offer sufficient support in the decision of adding punctuation marks. Before we describe the approach which yielded the highest accuracy we will introduce an n-gram based experiment which resulted in a very poor F-score of 0.56. Given a sentence of n tokens, similarly to the n-gram truecasing we tried to determine if a punctuation mark has to be inserted at any position inside the sentence from index 2 to n-1 (no probability of insertion was calculated for the beginning and the end of the sentence). The feature window was composed of 4 words centered on the position in which we want to determine the insertion probability. We used overlapping n-grams and computed the non-insertion probability as $P(w_{k-2}, w_{k-1}, w_k)P(w_{k-1}, w_k, w_{k+1})$ and the insertion probability as:

$P(w_{k-2}, w_{k-1}, PUNCT)P(PUNCT, w_k, w_{k+1})$.

Every time we calculated this probability for comma, the insertion probability was magnitudes higher than the non-insertion probability, resulting in the insertion of commas after almost every word in the utterance. Tweaking n-grams and manually adding rules did not yield much improvements in the insertion precision, thus we stopped this experiment and resorted to a different approach. We must note, that a LM build with higher order n-grams and based upon POS tags, rather than word forms, intuitively should produce better results. However, POS tagging on non-normalized text is not reliable and we preferred to employ a word form approach.

Neural inspired models have received an increasing interest from the research community. For us, an interesting

development was the unsupervised word embedding extraction method introduced by (Mikolov and Dean, 2013). Using large corpora, this method enables the automatic encoding of words into vector space. An important property is that semantically close words have close distance vectors, and this pre-processing method has produced remarkable results in tasks such as document classification (Xing et al., 2014; Kusner et al., 2015; Lai et al., 2015), sentiment analysis (Zhang et al., 2015), machine translation (sequence to sequence models) (Sutskever et al., 2014; Cho et al., 2014), prosodic modeling (Wang et al., 2015; Ding et al., 2015; Rallabandi et al., 2015; Rendel et al., 2016) etc.

Before we trained our classifier, we prepared our training data by running word2vec (Mikolov and Dean, 2013) on a large corpus and automatically extracting word embeddings. The vector size for the embeddings was set to 100. For the classification task we used a 3-layer network, with an input layer size of 600 units, a hidden layer size of 50 and an output layer size of 3. The input layer was fed with the word embeddings extracted from a window of 6 words. The window was slid from position 1 to position n-1 over the utterance. Sentence start and end were hardcoded as special input vectors which were used whenever the window exceeded the sentence boundaries. Unknown tokens were encoded using hardcoded vectors. The network was trained to output 3 states: (a) non-insertion, (b) comma insertion and (c) full stop. Our testing procedure was performed similarly to true casing. We kept aside 10% of the available data, which was stripped of punctuation marks. After this, we evaluated the system's capacity to reconstruct the original text. Individual performance values are shown in table 2. The system's F-score is 0.71.

Table 2 – Performance figures for punctuation restoration

Punctuation mark	Precision	Recall	F-score
Comma	0.92	0.59	0.72
Full stop	0.75	0.64	0.69
Mixed	0.87	0.60	0.71

4. Machine translation

For our machine translation approach we used the standard SMT Moses Decoder (Koehn et al., 2007) with a 3-gram language model constructed similarly to the approach described in section 2.1.
We trained the system on the in domain parallel data and the language model was built using the SRILM toolkit (Stolcke, 2002) surface-form, 5-gram, interpolated, using Knesser-Ney smoothing.

4.1. Translation equivalent inferred from monolingual corpora

In this experiment we attempted to see if adding translation equivalents from monolingual corpora is possible. In order to do this we performed the following steps:
Initially we used word2vec to obtain word embeddings in the source language.

Next we used GIZA++ to automatically align words from the source language with words from the destination language. We kept only 1-to-1 alignments in our data. We parsed the MOSES phrase table (trained for En-De and En-Fr) and we extracted n-grams from the source language. We computed the embeddings of these files by summing over the vectors computed earlier of each individual word.

We used a monolingual text corpus and we extracted unigrams, 2-grams, 3-grams, 4-grams and 5-grams. For every such n-gram we computed a vector space projection by summing over the centroid vectors of each individual word. The centroids were computed as a weighted sum over the dictionary translations obtained earlier. We used the translation probability as the weight.

We compared the obtained vector projections with every projection computed from the phrase table and we created new translation pairs if the distance between the vectors was below a given threshold. The threshold was heuristically chosen after a few experiments. We did not perform a grid-search to find a best-value threshold due to limited time and the fact that this was an experiment with unknown (better or worse) results.

```
to the installation  ||| à l' installation
that the same  ||| même que la
the same that  ||| même que la
```

Figure 1 - En-Fr example translation equivalents obtained by word embeddings

Using this method we added 70K translation equivalents to our phrase table. Whenever we encountered a translation equivalent already existing in the original phrase table, we skipped it – we added only new translation equivalents. This method increased our BLEU score with 0.5 on the validation set.

5. Results and future work

In what follows we show the accuracy figures for each individual track in which we participated. Our MT system scored a BLEU score of 0.296 for the En-Fr 2015 contrastive run and 0.269 for this year's track (see table 3). For the MSLT track we did not perform any domain adaptation of our translation system. Obviously, the score was extremely low (0.043) as shown in Table 4.
The ASR results are detailed in Table 5 for each individual speaker. Our system scored an average accuracy of 61.37% with the highest accuracy of 86.10% and the lowest accuracy of 37.91%.

Table 3 - Scoring of RACAI's MT submission

	BLEU	NIST	TER
TED.tst2015.MT_en-fr	0.296	6.844	0.526
TED.tst2016.MT_en-fr	0.269	6.636	0.549

Table 4 - Scoring of RACAI's MSLT submission

	BLEU	TER	BLEU c-i	TER c-i
MSLT.tst2016.en-fr.en	0.043	79.53	4.62	78.61

Table 5 - ASR Accuracy on the TED Talks

tst2016.EN.talktask.primary.ctm				
Talk	**#Snt**	**#Wrd**	**#Corr**	**Acc.**
talkid2227	41	904	736	81.42
talkid2284	110	1770	1524	86.10
talkid2286	88	1908	1338	70.13
talkid2309	53	835	341	40.84
talkid2313	28	546	207	37.91
talkid2319	85	1321	634	47.99
talkid2330	99	2061	1255	60.89
talkid2340	69	1135	691	60.88
talkid2341	108	1790	1052	58.77
talkid2344	132	2319	1503	64.81
talkid2357	133	1719	927	53.93
talkid2361	111	2281	1792	78.56
talkid2363	78	1624	741	45.63
alkidg3mcfu0b5hun	182	1699	874	51.44
talkidiyzklzutfr1d	148	1125	522	46.40
Overall	1465	23037	14137	61.37

We are very interested in spoken language translation systems and we intend to further focus our research on dialog oriented language models and improving the accuracy of our baseline ASR system. We intend to mainly focus on English and Romanian and we have already developed a processing pipeline that includes diacritic restoration (Romanian version) and text normalization for these two languages and we will further develop our systems until we reach an acceptable performance. In the field of ASR we intend to address neural inspired speech recognition methods as well as methods for combining alternative speech transcriptions based on the output of multiple systems and post-editing.

Acknowledgements

This research was supported by UEFISCDI grant PN-II-PT-PCCA-2013-4-0789 (ANVSIB project).

6. References

Ahn, K., Bos, J., Kor, D., Nissim, M., Webber, B. L., & Curran, J. R. (2005, November). Question Answering with QED at TREC 2005. In TREC.

Aw, A., Zhang, M., Xiao, J., & Su, J. (2006, July). A phrase-based statistical model for SMS text normalization.In Proceedings of the COLING/ACL on Main conference poster sessions (pp. 33-40).Association for Computational Linguistics.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., &Bengio, Y. (2014).Learning phrase representations using RNN encoder-decoder for statistical machine translation.arXiv preprint arXiv:1406.1078.

Ding, C., Xie, L., Yan, J., Zhang, W., & Liu, Y. (2015, December). Automatic prosody prediction for Chinese speech synthesis using BLSTM-RNN and embedding features. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (pp. 98-102).IEEE.

Eisele, A., & Chen, Y. (2010, May). MultiUN: A Multilingual Corpus from United Nation Documents. In LREC.

Gravano, A., Jansche, M., & Bacchiani, M. (2009, April). Restoring punctuation and capitalization in transcribed speech. In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4741-4744). IEEE.

Israel, R., Tetreault, J., &Chodorow, M. (2012, June).Correcting comma errors in learner essays, and restoring commas in newswire text. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 284-294). Association for Computational Linguistics.

Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015).From word embeddings to document distances.In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015) (pp. 957-966).

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, January).Recurrent Convolutional Neural Networks for Text Classification.In AAAI (pp. 2267-2273).

Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., ... & Wolf, P. (2003, April). The CMU SPHINX-4 speech recognition system. In IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong (Vol. 1, pp. 2-5).

Mikolov, T., & Dean, J. (2013).Distributed representations of words and phrases and their compositionality.Advances in neural information processing systems.

Rallabandi, S. K., Rallabandi, S. S., Bandi, P., &Gangashetty, S. V. (2015, December). Learning continuous representation of text for phone duration modeling in statistical parametric speech synthesis.In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (pp. 111-115).IEEE.

Rendel, A., Fernandez, R., Hoory, R., &Ramabhadran, B. (2016, March). Using continuous lexical embeddings to improve symbolic-prosody prediction in a text-to-speech front-end.In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5655-5659).IEEE.

Schlippe, T., Zhu, C., Gebhardt, J., & Schultz, T. (2010, September). Text normalization based on statistical machine translation and internet user support. In INTERSPEECH (pp. 1816-1819).

Sonmez, C., &Ozgur, A. (2014).A Graph-based Approach for Contextual Text Normalization.In EMNLP (pp. 313-324).

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., & Schlüter, P. (2013). Dgt-tm: A freely available translation memory in 22 languages. arXiv preprint arXiv:1309.5226.

Stolcke, A. (2002, September). SRILM-an extensible language modeling toolkit. In Interspeech (Vol. 2002, p. 2002).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

Wang, P., Qian, Y., Soong, F. K., He, L., & Zhao, H. (2015, April). Word embedding for recurrent neural network based tts synthesis. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4879-4883).IEEE.

Xing, C., Wang, D., Zhang, X., & Liu, C. (2014, December).Document classification with distributions of word vectors.In Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific (pp. 1-5).IEEE.

Zhang, D., Xu, H., Su, Z., &Xu, Y. (2015). Chinese comments sentiment classification based on word2vec and SVM perf. Expert Systems with Applications, 42(4), 1857-1863.