Evaluating LSTM Networks, HMM and WFST in Malay Part-of-Speech Tagging

Tien-Ping Tan[1], Bali Ranaivo-Malançon[2], Laurent Besacier[3], Yin-Lai Yeong[1], Keng Hoon Gan[1], and Enya Kong Tang[1]
[1]*School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.*
[2]*Faculty of Computer Science & Information Technology, Universiti Malaysia Sarawak, Sarawak, Malaysia.*
[3]*LIG, Université Grenoble Alpes, CNRS, Grenoble, France.*
tienping@usm.my

***Abstract*—Long short term memory (LSTM) networks have been gaining popularity in modeling sequential data such as phoneme recognition, speech translation, language modeling, speech synthesis, chatbot-like dialog systems and others. This paper investigates the attention-based encoder-decoder LSTM networks in Malay part-of-speech (POS) tagging when it is compared to weighted finite state transducer (WFST) and hidden Markov model (HMM). The attractiveness of LSTM networks is its strength in modeling long distance dependencies. Malay POS tagging is examined from two different conditions: with and without morphological information. The experiment results show that LSTM networks that are trained without any explicit morphological knowledge perform nearly equally with WFST but better than HMM approach that is trained with morphological information.**

***Index Terms*—Malay Part-Of-Speech Tagging; Recurrence Neural Network (RNN); Long Short Term Memory (LSTM) Networks, Sequence-To-Sequence Learning.**

I. Introduction

Recently, neural networks have been gaining popularity in the field of artificial intelligence. The advancements are due to the breakthrough in the algorithms that learn and recognize very complex patterns using deep layers of neural networks or commonly known as the deep neural networks (DNN) [1], and the introduction of different types of neural network such as convolutional neural network and recurrent neural network (RNN). For instance, convolutional neural networks, which are special type of feed-forward neural networks with two-dimensions networks, have shown tremendous accuracy in classifying images through local receptive fields, shared weights, pooling, from simple handwritten digit recognition to more complex face recognition. In the modeling of sequential patterns, such as phoneme recognition [2], automatic speech recognition [3][4], speech synthesis [5], speech translation [6], chatbot and many others, RNN or the more specialized type of RNN, the long short term memory (LSTM) networks have shown to be better than many of the traditional approaches.

This paper presents a comparative study of three methods to solve the problem of Malay part-of-speech (POS) tagging. These methods are LSTM networks, weighted finite state transducer (WFST) and hidden Markov model (HMM). The objective is to examine the performance of the current state of the art attention-based encoder-decoder LSTM networks while compared to WFST and HMM in POS tagging. POS tagging is a language processing task that assigned a POS tag (e.g., noun, verb, adjective, etc.) to each word in a sentence. Taking a different approach, in this study, the pairs of word/POS tag are not provided. Instead, the proposed model will learn the sequence-to-sequence mapping from the sequential data provided. The benefit of this approach is that, for certain languages without clear word boundary, the implicit word boundary knowledge is learnt from the data. The main challenge for the algorithm is to find the word alignment information from the data provided as illustrated in the examples in Table 1.

Table 1
Example Sentences and their POS

No	Malay Sentence	Meaning (English)	POS Annotation
1.	*pasaran buruh*	labor market	N N
2.	*kedua - dua benua*	both continents	NUM_CART N
3.	*cintaku*	my love	GEN_PRO N
4..	*kuala lumpur*	Kuala Lumpur	N

In addition, the examined approach must find the alignment between the word and its POS tag from the data, with the possibility that a word (a string separated by space) may map to more than one POS tag (example 3 in Table 1), or more than one word may map to a single POS tag (example 4 in Table 1).

II. Malay and POS Tagging

Malay is the official language used in Malaysia, Indonesia, Singapore, and Brunei. Malay is an agglutinative language. As such, new words can be created by adding one or several – less than three – affixes to a base word. The affixed can be the host of proclitic, enclitic and particle. Figure 1 shows the morphological structure of a Malay word [7].

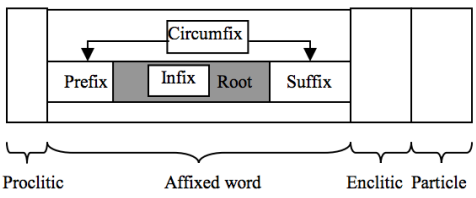

Figure 1: Morphological structure of Malay word [7]

The two proclitic (*ku*- 'I' and *kau*- 'you') and four enclitics (*-ku* 'me, my', *-kau* 'you, your', *-mu* 'you, your' and *-nya*

'him, her') assume different syntactic functions such as possessive, objective pronoun, subjective pronoun and definite article [7]. The three particles (*-kah*, *-tah* and *-lah*) are markers of interrogative, imperative and predicative sentences [7]. The nine prefixes, eight circumfixes and three suffixes carry a variety of meanings [7] as illustrated in Table 2 for the word *rata*. It is obvious that not all affixes can be added to any word.

Table 2
Possible affixations of the word "rata" in Malay

No	Word	Morphemes	Meaning
1.	rata	-	flat
2.	serata	*se-*	around, all over
3.	meratakan	*me-., -kan*	to flatten
4.	meratai	*me-, -i*	distribute
5.	perataan	*pe-, -an*	flattening
6.	keserataan	*ke-,se-, -an*	uniformity
7.	meratakannyalah	*me-, -kan, -nya, -lah*	flatten it (imperative)

The last decade has shown more and more works on Malay POS tagging. On one hand, researchers attempted to use rule-based approaches [8][9][10]. On the other side, machine learning techniques have been tested such as decision trees [11], k-nearest neighbor [11], maximum entropy model [12] and HMM [13]. The comparative study done by Xu and colleagues [12] using the same Malay corpora containing news articles showed that the current accuracy of existing Malay POS taggers varies between 46.67% (for a rule-based tagger) and 95.15% (for maximum entropy tagger [12]). In all the cited Malay POS tagging works, each word is tagged with one single POS tag. The tagging makes use of the affixation information to identify the POS tag.

III. Description of the Three Methods

A. *Weighted Finite State Transducer (WFST)*

A finite state transducer (FST) is a finite state machine that produces an output when it reads an input while traversing an edge in a state transition network. A WFST is an FST with weights (or Markov chain) on the edges. FST is useful for recognizing patterns that can be defined as a regular relation. Thus, in natural language processing, it is used in morphological analysis, grapheme to phoneme conversion, and as an alternative to the regular expression. For instance, in the morphological analysis, any two-level morphological rules can be implemented using an FST [14].

In POS tagging, the state transition network is used to map words to POS tags (Figure 2). The WFST POS tagging is a stochastic approach, where the idea is to select the tags with the highest joint probability of word and tag, $w_i t_i$.

$$T' = \arg\max P(w_1{:}t_1 , w_2{:}t_2 , ..., w_n{:}t_n) \quad (1)$$

where T' is sequence of tags $t_1, t_2, t_3 \ldots t_n$.

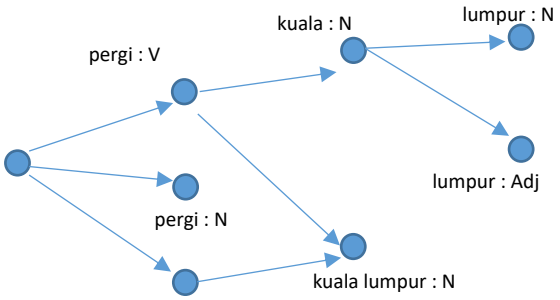

Figure 2: A snippet of the state transition diagram for the sentence "pergi kuala lumpur". Note: the transition weights are not included in the figure.

B. *Hidden Markov Model (HMM)*

A HMM is an extension of the Markov chain that consists of hidden states and observed states. HMM is one of the most frequent applied machine learning approach in part-of-speech tagging [15]. With HMM, the most probable tags given a sentence can be estimated as follow:

$$T' = \arg\max P(T \mid W) \quad (2)$$
$$= \arg\max P(T)\, P(W \mid T) \quad (3)$$

where T' is the sequence of most probable sequence of tag, T is the sequence of tags $t_i = t_1, t_2, \ldots, t_n$, and W is the sequence of words $w_i = w_1, w_2, \ldots, w_n$. P(T) can be modeled using POS n-gram. In situation where word and tag are explicitly associated, P(W|T) can be estimated using maximum likelihood estimation (MLE) with the following formula:

$$P(w_i|t_i) = \frac{C(w_i,t_i)}{C(t_i)} \quad (4)$$

But if the word and POS tag are not explicit, then the alignment between the word and POS tag should be carried out. The expectation maximization (EM) algorithm is often used for alignment. The intuition of the algorithm is to align word and POS tag that often seen together in the source target sequences. Initially, all alignments between source and target sequences are equally likely. After an iteration, source and target sequences that are often appearred together will have the likelihood improved.

Figure 3 shows an example HMM for the sentence *pergi Kuala Lumpur* 'go to Kuala Lumpur'.

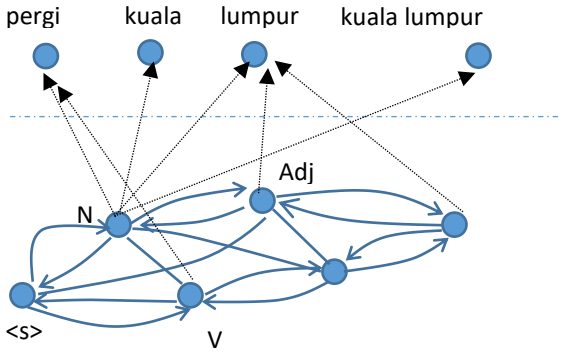

Figure 3: A snippet of the HMM for the sentence "pergi kuala lumpur". Note: the transition weights are not included in the figure.

Since there is no assumption on the ordering of the sequences in the source and target, the HMM approach can be applied in other situation such as machine translation.

C. LSTM Encoder-Decoder Sequence-to-Sequence Model

A recurrent neural network is a neural network with feedback loop to allow information to persist. An RNN can be thought as multiple copies of the same neuron passing information to its successor [16] as shown in Figure 4. The loop allows reasoning made in the previous neurons to affect the present neuron, which is important in sequential type of data mentioned in the introduction.

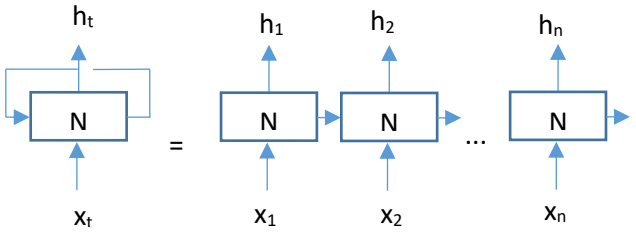

Figure 4: A Recurrent Neural Network. Note: x_i is input, h_i is output for a RNN

The limitation of the basic RNN is that it is not doing well in modeling the data that are a distance away in practice. LSTM networks are introduced to solve the long-term dependencies problem in RNN. In an LSTM network, there are gates that allow information to be forgotten and updated depending on the usefulness the information is.

LSTMs can form different types of networks. The encoder-decoder networks have been demonstrated to be very good in sequence to sequence modeling [17][18] (see Figure 5). Given a source sequence (e.g. words), the encoder will encode the input as a vector and passes it to the decoder. The decoder will generate output from the vector passed from the encoder until a special end of sentence tag is reached. An attention-based encoder-decoder LSTM network will allow certain part of the source sequence to attend or focus on certain part of the target sequence during training and decoding, instead of the whole sentence encode as a single vector. In another word, the attention values tell the strength of the alignment between a combination of input words and output words, allowing more context specific encoding and decoding [19].

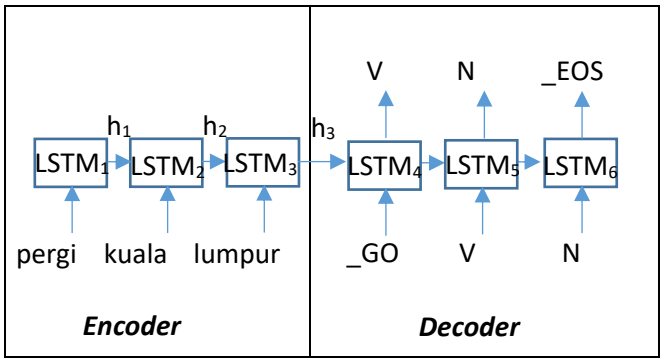

Figure 5: LSTM Encoder-Decoder. Note: the words will be converted to embedding vectors before input to the LSTM.

IV. Experiments and Results

The Malay POS annotated text used in the experiments consists of 423,767 sentences, which were tagged using 36 POS tags [20]. Contrary to the usual norm where each word is assigned a POS tag, the word in the tagged sentences is not assign a tag explicitly. Instead, the tags are implicitly assigned in sequence to one or more words, just like in a parallel text in machine translation. Thus, the POS training algorithm must learn the word(s)/POS alignments from the "word/POS parallel text" as shown in Table 1. From the total tagged sentences, about 400k sentences were used for training, and the remaining 2 * 10k sentences was used for testing and development respectively.

We examine two different test conditions. In the first test case (TC1), all sentences were only normalized with some simple preprocessing steps, for instance, the numbers were normalized using regular expressions, and in another test case (TC2), the sentences in the first test case were further processed based on their morphological information. We used the Malay morphological analyzer proposed by Ranaivo-Malançon and colleagues [21] for the morphological analysis.

Original sentence: *telah menerima sejumlah 5.9 juta pelawat dan 480 pertanyaan sejak pelancarannya pada tahun 2003*

Test Case 1 (TC1): *telah menerima sejumlah [REAL] juta pelawat dan [DIGIT] pertanyaan sejak pelancarannya pada tahun [DIGIT]*

Test Case 2 (TC2): *telah me+ terima se+ jumlah [REAL] juta pe+ lawat dan [DIGIT] per+ tanya +an sejak pe+ lancar +an +nya pada tahun [DIGIT]*

The rule-based morphological analyzer will produce more than one segmentations. In most cases, it is possible to manually select the valid segmentation from few possible segmentations.

mengabui -> me+ kabui (valid segmentation)
-> meng+ abu + i (invalid segmentation)

However, certain words present several valid segmentations. The right segmentation will depend on the context of the word in the sentence. The most notable one is the word "mereka":

mereka -> mereka (meaning: they)
-> me+ reka (meaning: to design)

Since segmenting a word based on its context in the sentence requires a lot of time and resources, we manually selected only the most common segmentation without considering the context.

The size of the vocabulary for TC1 in the training set is 29,004, and we set the words that appear only once in the training data to _UNK (unknown) tag. We ended up with 25,000 words in the vocabulary including the _UNK. This means that about 4,000 words that appear only once were set to _UNK. On the other hand, for TC2, there were 15,328 total words in the vocabulary, and we set the words that appear only once to _UNK, and ending up with a vocabulary with 13,000 words. The accuracy of the Malay POS tagging was evaluated using the sclite from NIST scoring toolkit.

A. WFST Approach Using Phonetisaurus

For testing the WFST approach, we used Phonetisaurus [22]. Phonetisaurus is a tool first proposed for learning

grapheme to phoneme mapping rules in pronunciation modeling. The attractiveness of Phonetisaurus is that the alignment algorithm proposed in it uses a modified EM approach which can learn the source to target many-to-many alignments. The joint probability of the source-target sequence can then be modeled with an n-gram language model or finite state machine. Here, we applied Phonetisaurus to find the words/ POS tags alignments. The aligned joint label pairs of word/POS tag obtained using the EM algorithm in Phonetisaurus are then used to build an n-gram language model. We applied SRI language modeling toolkit on the training set to learn the joint probability for word: POS n-gram language models with different orders using Kneser-Ney discounting strategy, and then converted the n-gram language model to a FST network. The following Table 2 shows the POS tagging error rates carried out on the test set obtained using 4-grams, 5-grams, and 6-grams.

Table 2
POS Tagging Error Rate with WFST (Phonetisaurus)

	4-grams	5-grams	6-grams
TC1	*18.7%*	18.6%	18.6%
TC2	*11.8%*	11.5%	11.4%

The results show that Phonetisaurus with higher order 6-grams gives the best result with 18.6% tagging errors in TC1 and 11.4% in TC2. The experiment also shows that including the morphological information in TC2 improve the alignments produced by Phonetisaurus, reducing the error rate more than 7%.

B. HMM Approach Using Moses

Next, we evaluated the HMM approach by using Moses MT toolkit [23]. We select Moses due to the state-of-the-art MT results, even though Moses is not a pure HMM approach. The maximum length of a phrase in Moses for TC1 is set to 4, and the maximum length of a phrase for TC2 is set to 14. The following is the results obtained for the test set:

Table 3
POS Tagging Error Rate with HMM (Moses MT)

	3-grams	4-grams
TC1	*16.8%*	16.7%
TC2	*14.8%*	14.6%

C. Attention-based Encoder-Decoder LSTM Networks

In this evaluation, we used the multi-layered bidirectional attention-based encoder-decoder LSTM networks implemented by Berard et al. [6] using Google's Tensorflow framework (https://www.tensorflow.org/) to test the Malay POS tagging. The LSTM models were trained using the training set and configure with the development set. We tested different sizes of word embedding vectors, and we found that 256 is the most optimum size for Malay POS tagging. Besides that, we also tried using pretrained word embedding vectors with word2vec algorithm, but it did not improve the results. The optimizer used to train the LSTM networks was set to Stochastic Gradient Descent (SGD) with learning rate of 0.5 and decay 0.99. Beam size for decoding was set to 4. Table 4 shows the error rate of the POS tagging carried out on the test set using 3 layers of RNN with varying size of LSTM cells in second to forth columns, and the last column shows the result using 4 ensemble LSTM networks.

Table 4
POS Tagging Error Rate with Attention-based Encoder-Decoder LSTM Networks

	64	128	256	256 (Ensemble)
TC1	*17.1%*	16.6%	16.2%	11.8%
TC2	*17.6%*	17.1%	16.6%	12.5%

The results show that increasing the size of LSTM cell will reduce the POS tagging error rate. The best result from LSTM networks is using 256 size cells, where the error rate stands at 16.2% for TC1 and 16.6% for TC2. This result for TC1 is slightly better than the other approaches reported in WFST and HMM earlier. When we combined 4 of the best LSTM models using ensemble approach, the error rate in TC1 drops to 11.8%. This result from TC1 is comparable to the best result produced by WFST (which is 11.4%) with morphological information (TC2)! This result is very intriguing because it demonstrates the power of LSTM networks in capturing morphological knowledge of the data even when this knowledge is not even provided.

However, when we just compared TC2 results (not ensemble) in all the approaches, LSTM networks do not give the best results. It seems that the LSTM networks is not able to associate between the morphological information provided and POS tag very well during training. This might be due to the limitation of the "attention" in the LSTM networks. We also tried to use POS language model built for HMM to restore the results we got from TC1 and TC2, but it does not give any improvement to the results.

V. CONCLUSION

The WFST produces a more accurate POS tagging compared to HMM and encoder-decoder LSTM networks when morphological information is provided. However, when an ensemble of LSTM networks which are trained without using morphological information, we see that the results is nearly equivalent to what we get with WFST that are trained with morphological knowledge. This shows that LSTM networks can capture the morphological knowledge for a language. This has tremendous benefit especially to be used on languages that we do not have much linguistic studies on.

However, in term of training the models, WFST is the fastest to train and run. Qualitatively, a WFST takes only few minutes to run about 400+k training. This is followed by HMM (3 hours), and subsequently followed by LSTM networks (more than 8 hours). In term of decoding time, WFST is the fastest, followed by HMM and then LSTM. Nevertheless, all can decode in real time speed. As for the number of parameters used in training, WFST is the simplest to setup to run, followed by HMM and LSTM.

ACKNOWLEDGMENT

This work is supported by Fundamental Research grant (FRGS) from Ministry of Higher Education of Malaysia: 203.PKOMP.6711536

REFERENCES

[1] M. Nielsen, "Neural networks and deep learning," Online: http://neuralnetworksanddeeplearning.com/, 2017.

[2] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, Vancouver, Canada, 2013, pp. 6645–6649.

[3] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *in Proc. INTERSPEECH*, Singapore, 2014, pp.338-342.

[4] M. Sundermeyer, R. Schluter, and H. Ney, "Lstm neural networks for language modeling." *in Proc. INTERSPEECH*, Portland, 2012, pp. 194–197.

[5] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," *in Proc. ICASSP*, Brisbane, 2015, pp. 4470–4474.

[6] A. Bérard, O. Pietquin, L. Besacier, and C. Servan, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *Conf. on Neural Information Processing Systems (NIPS)*, Barcelona, 2016, pp. 1–5.

[7] B. Ranaivo-Malançon, "Computational analysis of affixed words in Malay language," in *ISMIL*, Penang, 2004, pp. 1-11.

[8] G. Knowles and Z. M. Don, *Word Class in Malay: A Corpus-Based Approach*, Kuala Lumpur: Dewan Bahasa dan Pustaka, 2006.

[9] R. Alfred, A. Mujat, and J. H. Obit, "A ruled-based part of speech (RPOS) tagger for Malay text articles," in *Conf. on Intelligent Information and Database Systems*, Kuala Lumpur, 2013, pp. 50-59.

[10] M. P. Hamzah, B. S. Kamaruddin and S. F. Na'imah, "Part of speech tagger for Malay language based on words morphology," in *Int. Sym. on Research in Innovation and Sustainability*, Melaka, 2014, pp. 1409-1502.

[11] J. A. Bakar, K. Omar, M. F. Nasrudin and M. Z. Murah, "Morphology analysis in Malay pos prediction," in *Proc. of the Int. Conf. on Artificial Intelligence in Computer Science and ICT*, Langkawi, 2013, pp. 112-119.

[12] B. M. X. Chu., M. Lubani, K. P. Liew, K. Bouzekri, R. Mahmud, and D. Lukose, "Benchmarking mi-pos: Malay part-of-speech tagger," *International Journal of Knowledge Engineering*, vol. 2, no. 3, pp. 115-121, 2016.

[13] M. Hassan, N. Omar, and M. J. A. Aziz, "Statistical Malay part-of-speech (POS) tagger using hidden Markov approach," in *Conf. on Semantic Technology and Information Retrieval*, Putrajaya, 2011, pp. 231-236.

[14] P. M. Nugues, *An Introduction to Language Processing in Perl and Prolog*, New York: Springer, 2010, pp.133-144.

[15] D. Jurafsky and H. James, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech*, New Jersey: Prentice Hall, 2000.

[16] C. Olah, "Understanding Lstm networks," Online: http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[17] K. Cho, B. Van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Q. V. Le, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, 2014, pp. 1724–1734.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, pp. 3104-3112, 2014.

[19] D. Britz, "Attention and memory in deep learning and nlp," Online: http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/, 2016.

[20] H. N. Lim, H. H. Ye, C. K. Lim and E. K. Tang, "Adapting an existing example-based machine translation (ebmt) system for new language pairs based on an pptimized bilingual knowledge bank (bkb)," *Int. Conf. on Translation*, Kuala Lumpur, 2007, pp. 399-406.

[21] B. Ranaivo-Malançon, C. C. Chua, P. K. Ng, "Identifying and classifying unknown words in Malay texts," in *Int. Sym. on Natural Language Processing*, Pattaya, 2007, pp. 493-498.

[22] J. R. Novak, N. Minematsu, K. Hirose, "Wfst-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding," *in Int. Workshop on Finite State Methods and Natural Language Processing*, Donostia–San Sebastia, 2012, pp. 45-49.

[23] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran and R. Zens, "Moses: open source toolkit for statistical machine translation," in *Proc. of the 45th annual meeting of the ACL*, Prague, 2007, pp. 177-180.