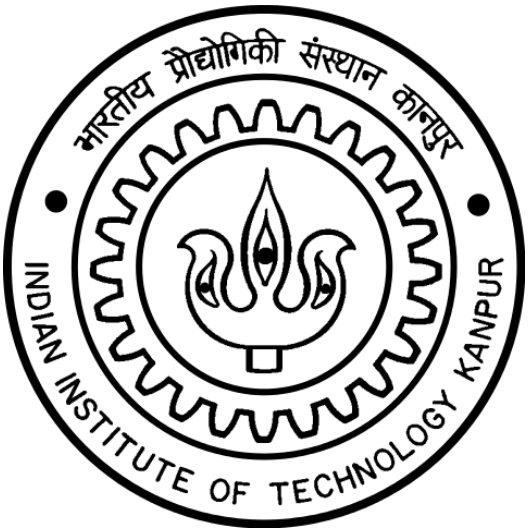
CS671A
Natural Language Processing

Hindi ↔ English Parallel Corpus Generation from Comparable Corpora for Neural Machine Translation



Ekansh Gupta (12252)
Rohit Gupta (15111037)
Mentored by: Prof. Amitabha Mukerjee

Index

Abstract

Neural Machine Translation (NMT) is a new approach to the well-studied task of machine translation, which has significant advantages over traditional approaches in terms of reduced model size, and better performance. NMT models require a parallel corpus of significant size to be trained, which is lacking for the Hindi ↔ English language pair. However, significant amounts of comparable corpora are available. The aim of this project is to develop a technique to produce a high quality parallel sentence-aligned corpus from existing subject/document-aligned comparable corpora (Wikipedia).

Introduction

Machine translation

Traditionally, Statistical Machine Translation (SMT) systems have framed the translation problem as follows: [Ref: Cho's Tutorial]

Maximize log(p(f|e))
log(p(f|e)) = log(p(e|f)) + log(p(f)) + C

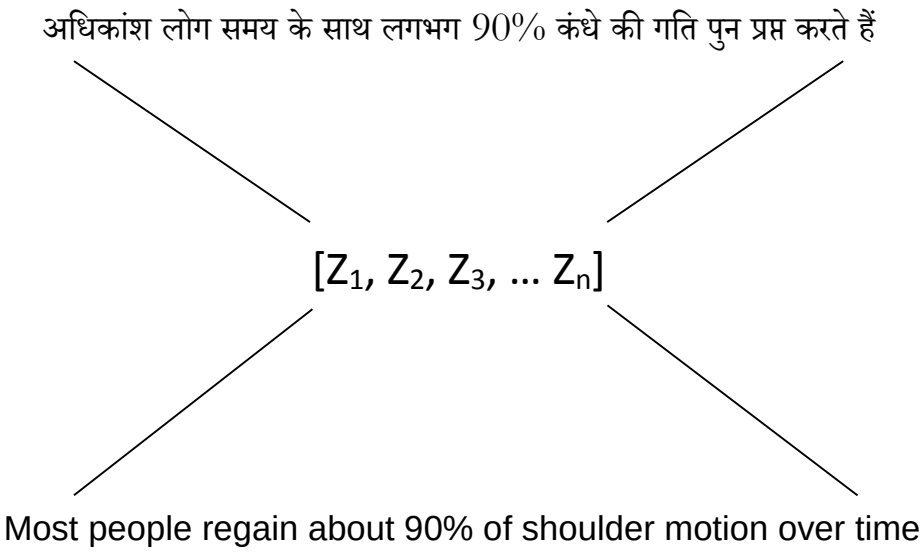
- ★ p(f|e): probability of foreign language sentence f given a native language sentence e
- ★ p(e|f): translation model
- ★ p(f): (target) language model
- ★ C: constant

This breaks up the problem into different parts, where a lots of features are used as parts of the model based on the researcher's intuition and experience. The separate parts (language model, translation model, etc. are all trained separately).

In the past few years, Neural Networks have had a lot of success in Computer Vision, Speech Recognition and now also in Natural Language Processing. The initial foray of Neural Networks in translation was in language models (p(f)). [Ref: Yoshua Bengio et al's *A Neural Probabilistic Language Model*]. Since then neural networks have also been used in translation models (p(e|f)) [Ref: Devlin et al *Fast and Robust Neural Network Joint Models for Statistical Machine Translation*] and then to re-rank suggestions generated by traditional SMT systems. The natural next step is end-to-end neural machine translation.

Neural Machine Translation (NMT) looks at translation as an end-to-end trainable supervised machine learning problem:

The following figure shows this scheme (the sentence pair shown in the figure by the way, was generated by our system from a topic-aligned Wikipedia article pair)



Note the salient properties of this task:

1. It maps a variable length input to a variable length output.
2. It is sensitive to word order.
3. It is a probabilistic many-to-many mapping.

Types of corpora

Large collections of parallel texts are called parallel corpora. Alignments of parallel corpora at sentence level are a prerequisite for many areas of linguistic research. During translation, sentences can be split, merged, deleted, inserted or reordered by the translator. This makes alignment a non-trivial task.

A comparable corpus is built from non-sentence-aligned and untranslated bilingual documents, but the documents are topic-aligned.

Quite a few machine translators use SMT (Statistical Machine Translation). Neural machine translation is a new approach to machine translation, where we train a neural network to achieve translation. This is a different from existing (phrase-based) machine translation approaches, where a translation system consists of many components which are optimized separately. Here we use RNN to train a weak translator which then is used to generate parallel corpora from comparable corpora.

Theory

To put it systematically, a **parallel corpus** is a corpus that contains a collection of original texts in language L_1 and their translations into a set of languages L_2 ... L_n. In most cases, parallel corpora contain data from only two languages.

Parallel corpora can be bilingual or multilingual, i.e. they consist of texts of two or more languages.
Closely related to parallel corpora are 'comparable corpora', which consists of texts from two or more languages which are similar in genre, topic, register etc. without, however, containing the same content.

A host of language processing utilities (including machine translation) rely directly on the availability of parallel corpora. Translation, in fact needs a huge amount of parallel corpora.

Very large parallel English-Hindi corpora are however, unavailable. On the other hand, topic and/or document aligned comparable corpora like Wikipedia, Proceedings of the Indian Parliament, news articles, etc are available much more readily. Parallel sentences may also be mined from comparable corpora such as news stories written on the same topic in different languages

Neural Machine Translation

A new approach for statistical machine translation based purely on neural networks has recently been proposed (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014). This new approach is inspired by the recent trend of deep representational learning. All the neural network models used in (Sutskever et al., 2014; Cho et al., 2014) consist of an encoder and a decoder. The encoder extracts a fixed-length vector representation from a variable length input sentence, and from this representation the decoder generates a correct, variable-length target translation.

Advantages of Neural Machine Translation

- Require only a fraction of the memory needed by traditional statistical machine translation (SMT) models
- Deep Neural Nets outperform previous state of the art methods on shorter sentences assuming availability of large parallel corpora
- For longer sentences NMT approaches can be combined with word-alignment approach to address the rare-word problem

Preliminary: Recurrent Neural Networks

A recurrent neural network (RNN) is a neural network that consists of a hidden state **h** and an optional output $\boldsymbol{y}$ which operates on a variable length sequence $\boldsymbol{x} = (x_1, \ldots\ldots, x_T)$. At each time step *t*, the hidden state h_t of the RNN is updated by

$$h_t = f(h_{t-1}, x_t)$$

The Recurrent Neural Network (RNN) is a natural generalization of feedforward neural networks to sequences. Given a sequence of inputs $(x_1, \ldots\ldots, x_T)$ a standard RNN computes a sequence of outputs $(y_1, \ldots\ldots, y_T)$ by iterating the following equation:

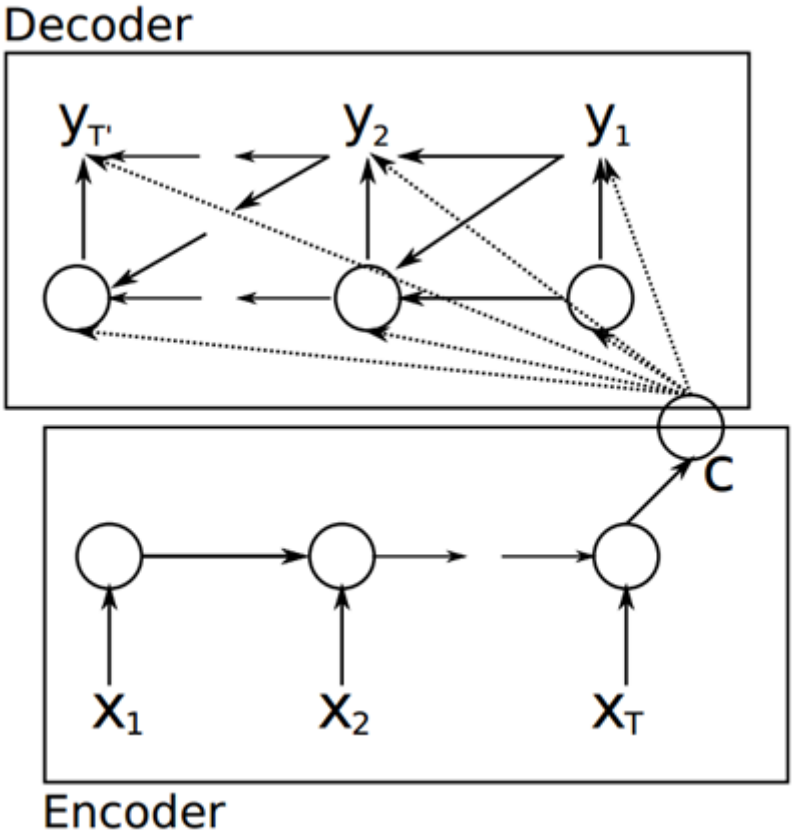
$$h_t = sigm(W^{hx} x_t + W^{hh} h_{t-1})$$

$$y_t = W^{yh} h_t$$

The RNN can easily map sequences to sequences whenever the alignment between the inputs the outputs is known ahead of time. However, it is not clear (or rather was, as there have been few techniques to tackle this. RNN encoder-decoder, one of these techniques, is what we are about to discuss next) how to apply an RNN to problems whose input and the output sequences have different lengths with complicated and non-monotonic relationships.

RNN Encoder-Decoder

To apply RNN to problems whose input and output have different lengths, we use a novel architecture proposed by [Cho et al] that learns to encode a variable length sequence into a fixed-length vector representation and to decode a given fixed-length vector representation back into a variable-length sequence. From a probabilistic perspective, this new model is a general method to learn the conditional distribution over a variable-length sequence conditioned on yet another variable-length sequence.



The RNN encoder-decoder architecture proposed by Cho et al.

Sentence Aligners

Lastly, we come to a sentence aligner. Sentence aligners are used for creating a one-to-one matching between sentences in two texts. They often use a similarity metric to find correspondence between sentences and score them based on that. Sentences that pass a similarity threshold are then assumed to be similar and can be used for further processing. Aligners are important for generation of parallel corpora from topic aligned texts.

Methodology

In addition to the RNN encoder-decoder, we used a new type of hidden unit that has been motivated by the LSTM unit but is much simpler to compute and implement.
Let us describe how the activation of the j^{th} hidden unit is computed. First, the reset gate r_j is computed by:

$$r_j = \sigma([W_r x]_j + [U_r h_{t-1}]_j)$$

Where σ is the logistic sigmoid function, and $[\dots.]_j$ denotes the j^{th} element of a vector. $\boldsymbol{x}$ and $\boldsymbol{h_{t-1}}$ are the input and the previous hidden state, respectively. $\boldsymbol{W_r}$ and $\boldsymbol{U_r}$ are weight matrices which are learned. Similarly, the update gate z_j is computed by:

$$z_j = \sigma\big([W_z x]_j + [U_z h_{t-1}]_j\big)$$

The actual activation of the proposed unit h_j is then computed by:

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\check{\mathrm{h}}_j^t$$

Where

$$\check{\mathrm{h}}_j^t = \phi([Wx]_j + [U(r \odot h_{t-1})]$$

Scraping

Wikipedia was used as a resource for collection of comparable corpora. Scraping was done using the MediaWiki API to fetch Hindi and English articles and a python library, BeautifulSoup4 was used to extract meaningful content from the HTTP response

Details: MediaWiki returns an XML response which contains all pages that it can find beginning with a particular prefix. From our list of Hindi consonants and vowels we iteratively downloaded Hindi Wikipedia articles. After we download a page, we search through its HTML to find if it had a corresponding English equivalent. In such a situation we scraped both article pairs and used BeautifulSoup4 to extract coherent text from the article.

Thereafter, the text was cleaned to remove unwanted characters and was stored in a line by line format. Due to the restricted speed of fetching articles by the API we managed to scrape about 600 article pairs.

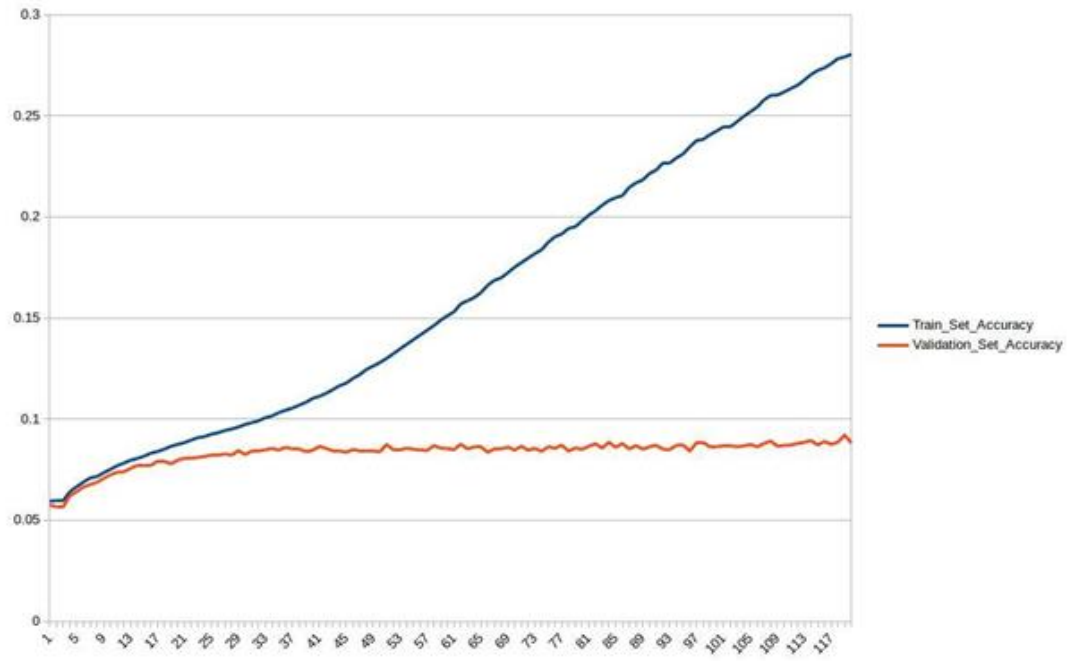
Pre-Processing

The words in each sentence were encoded using one-hot encoding for top-N words chosen from large monolingual corpora for each language (N=5000). Out of Vocabulary (OOV) words represented by <UNK>. Monolingual corpora used for choosing vocabulary range was obtained from HelioHost. To save on memory, sentences were limited to a maximum of 15 words.

RNN Encoder ↔ Decoder Model

The Keras Library was used to implement the RNN encoder-decoder translation model. The model encoded each input sentence (variable length) into a fixed-sized vector of length 512. The vocabulary size used was 5000 for both Hindi and English.

The translator was trained using 25000 sentence pairs on a GPU system with 4GB memory. The train-test split was 10%-90%.

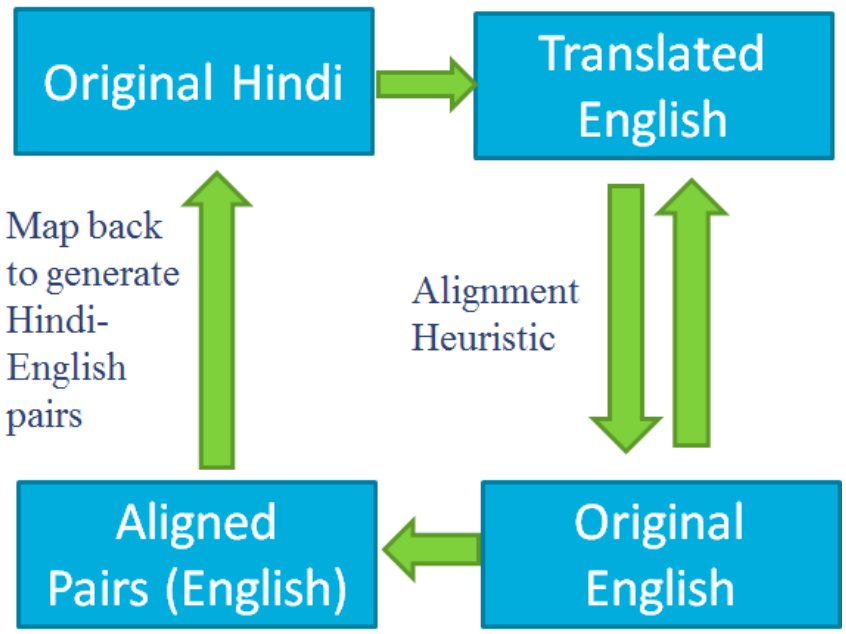


The steadily increasing training accuracy but low validation accuracy is a result of over-fitting due to a relatively small set of sentence pairs which was limited by the GPU memory and time. As a result, to generate parallel corpus, a python library, TextBlob, which is a weak alternative for translators was used. The RNN translator output is shown in the results section.

Sentence Alignment

We built a heuristic for aligning sentences between native English and weakly translated English, it used the token sort ratio metric (number of similar words).

A threshold was set above which if sentences were found similar they would be included in the matched list. We also set a parameter that calculated the fraction of length a sentence was compared to other sentences to avoid the matching of sentences with disproportionate difference in lengths.



The summary therefore looked like this:

- Training weak translator using limited parallel corpus
- Weak translator and aligning heuristic used to create additional parallel corpus
- Neural translator re-trained on generated bigger parallel corpus

Results

Neural Translation Results:

To improve the translation results on systems with low memory, we trained our model again this time using 50000 sentences, feeding a batch of 12500 sentence pairs iteratively over 4 cycles of 40 epochs each.

The results were better than the previously trained model on 25000 sentence pairs. As opposed to our prior results where the translator gave an output consisting mostly of <UNK>, it showed some promise in learning context this time around. Few of the translated sentences are shown below. Note that the character (ं) signifies the end of sentence in Hindi.

यह प्रतिशत भारत में हिन्दुओं प्रतिशत से अधिक है ं ं ं ं ं ं
this percentage is hindus hindus greater greater greater is india india india of of <UNK>

कड़ी नहीं खोल सका ं ं ं ं ं ं ं ं ं ं ं
could not open link link file file a a a a a a a a

वेद बहुत ही जटिल तथा **<UNK> <UNK>** में लिखे गए हैं ं ं ं ं
vedas are and <UNK> complex complex complex brief brief complex complex art art art art

रॉयल **<UNK>** राष्ट्रीय उद्यान ं ं ं ं ं ं ं ं ं ं ं
royal royal national park <UNK> <UNK> <UNK> centre centre centre centre centre of of centre

समन्वयन त्रुटि समन्वयन **<UNK>** अपडेट करें ं ं ं ं ं ं ं ं ं
sync error update sync passphrase passphrase sync sync sync sync sync sync sync try try

खोज विंडो की ऊँचाई को सहेजा ं ं ं ं ं ं ं ं ं
saved of position search search window window window window window window window window window window window

Parallel corpus generation results

The 600 article pairs were used to generate sentence pairs. About 1500 matching sentence pairs were generated after putting a reasonably high threshold in the aligner parameters for finding similarity. In order to evaluate the quality of the generated corpora we sample 100 of those pairs and got them rated by 4 different human evaluators (other than the two of us, of course). People rated them on a 3 point scale which was:

Perfect: When a sentence pair is fit to be included in Hindi-English parallel corpora
Acceptable: Contains few anomalies and would not be preferred for inclusion in parallel corpora
Rejected: Contains a lot of noise and not fit as a pair

Since we cannot be completely sure if 100 samples fairly represent the large pool that they were mined from, we include the sampling error with 95% confidence interval. The results were:

Perfect	**77%+7.9%**
Acceptable	**15%+6.7%**
Rejected	**8%+5.09%**

Some of the generated sentence pairs with their respective labels are:

Perfect

तंत्रिका में हड्डियों का द्रव्यमान ३० वर्ष की आयु के लगभग अपने अधिकतम घनत्व पर पहुँचती है
The bone mass in the skeleton reaches maximum density around age 30

यह टेंडन और मस्पेशियाँ जो प्रगंडिका के सर को विवर के ग्लेनोइड में पकड़ के रखता है
It is composed of the tendons and muscles that hold the head of the humerus in the glenoid cavity

---

फ्रोजेन शोल्डर के कारण होने वाला दर्द आम तौर मंद या पीड़ादायक होता है
Pain due to frozen shoulder is usually dull or aching

जर्मन विभिन्न तरह की संरचनाओं का निर्माण कर सकते हैं
The Germans can build a variety of structures

Perfect (continued)

एबल कंपनी को पहाड़ी पर कब्जा करने का दायित्व दिया जाता है
Able Company is assigned to take the hill

लेकिन कोई भी संचिका किसी नामस्थान में शून्य एक या अधिक नामों से संदर्भित की जा सकती है
However any file may be represented within any namespace by zero one or more names

---

इन अन्य फ़ोल्डरों को उपफ़ोल्डर कहते हैं
These other folders are referred to as subfolders

हर प्रणाली के अपने फ़ायदे नुकसान हैं
Each system has its own advantages and disadvantages

Average

न्यूनतम स्तर पर कई आधुनिक प्रचालन प्रणालियाँ संचिकाओं को केवल एकआयामी बाइटों की शृंखला ही मानती हैं
On most modern operating systems files are organized into one dimensional arrays of bytes

---

कई आधुनिक संगणक प्रणालियाँ संचिकाओं को नुकसान पहुँचने से रोकने के तरीके प्रदान करती है
Many modern computer systems provide methods for protecting files against accidental and deliberate damage

---

यह बेसर के समान है जिसका जन्म एक घोड़े और गधी के मिलन के परिणामस्वरूप होता है
A mule is the offspring of a male donkey and a female horse

Rejected

इस से कैटरीना की सफल फिल्मों की झड़ी जारी रही हालांकि उनको अपने अभिनय के लिए मिश्रित समीक्षा मिली
However she received mixed reviews for her performance

इनमें शामिल हैं प्राचीन ग्रीस की एंटिकिथेरा प्रक्रिया और एस्ट्रॉलैब जिन्हें आम तौर पर सबसे प्रारंभिक ज्ञात यांत्रिक एनालॉग कंप्यूटर माना जाता है
The Antikythera mechanism is believed to be the earliest mechanical analog computer according to Derek J

Discussion and future work

Based on the progress of the project at this juncture, some directions for future work include:

1. Pace of parallel corpus generation has been limited due to the slow pace of scraping, we could do better by obtaining dumps of Hindi/English Wikipedia. Additionally other sources of comparable corpora like proceedings of the Indian parliament and bilingual publications could be also be added.
2. The Neural Translation model has not achieved its potential. Significant performance gains can be had by using additional training data with the existing model.
3. In addition to using more data, using a bigger vocabulary is also important for achieving a high quality translator.
4. Improvements to the model: We could tune the model further by experimenting with improvements like use of bi-directional RNN, optimizing size of hidden state, jointly learning to align and translate, dealing with the rare word problem etc.
5. Studying the nature of the embedding produced (the internal hidden state) and the model would also help us better understand the task. (Similar to how google created Deep Dream to better understand ConvNets) This understanding could also help us leverage these embedding and the model for other tasks.

References

Papers

1. Kalchbrenner and Blunsom, Recurrent Continuous Translation Models, 2013
2. Sutskever et al, Sequence to Sequence Learning with Neural Networks, 2014
3. Cho et al, On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, 2014
4. Hermann and Blunsom, Multilingual Distributed Representations without Word Alignment, 2014
5. Cho et al, Neural Machine Translation by jointly learning to align and translate, ICLR 2015
6. Wolk and Marasek, Building subject aligned comparable corpora and mining it for truly parallel sentence pairs, 2014

Resources

1. **Heliohost**: http://corpora.heliohost.org/
2. **MediaWiki API**: https://www.mediawiki.org/wiki/API:Main_page
3. **BeautifulSoup4**: http://www.crummy.com/software/BeautifulSoup/
4. **Hindi-English Parallel Corpus**: https://lindat.mff.cuni.cz/repository/xmlui/handle/11858/00-097C-0000-0001-BD17-1
5. **Keras Deep Learning Library**: http://keras.io/