

Teaching a Massive Open Online Course on Natural Language Processing

Ekaterina Artemova^{1,2*}, Murat Apishev¹, Veronika Sarkisyan¹,
Sergey Aksenov¹, Denis Kirjanov³, Oleg Serikov^{1,4}

¹ HSE University, Moscow, Russia

² Huawei Noah's Ark lab, Moscow, Russia

³ SberDevices, Sberbank, Moscow, Russia

⁴ DeepPavlov, MIPT, Dolgoprudny, Russia

<https://openedu.ru/course/hse/TEXT/>

Abstract

This paper presents a new Massive Open Online Course on Natural Language Processing, targeted at non-English speaking students. The course lasts 12 weeks; every week consists of lectures, practical sessions, and quiz assignments. Three weeks out of 12 are followed by Kaggle-style coding assignments.

Our course intends to serve multiple purposes: (i) familiarize students with the core concepts and methods in NLP, such as language modeling or word or sentence representations, (ii) show that recent advances, including pre-trained Transformer-based models, are built upon these concepts; (iii) introduce architectures for most demanded real-life applications, (iv) develop practical skills to process texts in multiple languages. The course was prepared and recorded during 2020, launched by the end of the year, and in early 2021 has received positive feedback.

1 Introduction

The vast majority of recently developed online courses on Artificial Intelligence (AI), Natural Language Processing (NLP) included, are oriented towards English-speaking audiences. In non-English speaking countries, such courses' audience is unfortunately quite limited, mainly due to the language barrier. Students, who are not fluent in English, find it difficult to cope with language issues and study simultaneously. Thus the students face serious learning difficulties and lack of motivation to complete the online course. While creating new online courses in languages other than English seems redundant and unprofitable, there are multiple reasons to support it. First, students may find it easier to comprehend new concepts and problems in their native language. Secondly, it may be easier to build a strong online learning community if students can express themselves fluently. Finally, and

more specifically to NLP, an NLP course aimed at building practical skills should include language-specific tools and applications. Knowing how to use tools for English is essential to understand the core principles of the NLP pipeline. However, it is of little use if the students work on real-life applications in the non-English industry.

In this paper, we present an overview of an online course aimed at Russian-speaking students. This course was developed and run for the first time in 2020, achieving positive feedback. Our course is a part of the HSE university's online specialization on AI and is built upon previous courses in the specialization, which introduced core concepts in calculus, probability theory, and programming in Python. Outside of the specialization, the course can be used for additional training of students majoring in computer science or software engineering and others who fulfill prerequisites.

The main contributions of this paper are:

- We present the syllabus of a recent wide-scope massive open online course on NLP, aimed at a broad audience;
- We describe methodological choices made for teaching NLP to non-English speaking students;
- In this course, we combine recent deep learning trends with other best practices, such as topic modeling.

The remainder of the paper is organized as follows: Section 2 introduces methodological choices made for the course design. Section 3 presents the course structure and topics in more details. Section 4 lists home works. Section 5 describes the hosting platform and its functionality.

2 Course overview

The course presented in this paper is split into two main parts, six weeks each, which cover (i) core

Corresponding author, email: elartemova@hse.ru

NLP concepts and approaches and (ii) main applications and more sophisticated problem formulations. The first six weeks’ main goal is to present different word and sentence representation methods, starting from bag-of-words and moving to word and sentence embeddings, reaching contextualized word embeddings and pre-trained language models. Simultaneously we introduce basic problem definitions: text classification, sequence labeling, and sequence-to-sequence transformation. The first part of the course roughly follows Yoav Goldberg’s textbook (Goldberg, 2017), albeit we extend it with pre-training approaches and recent Transformer-based architectures.

The second part of the course introduces BERT-based models and such NLP applications as question answering, text summarization, and information extraction. This part adopts some of the explanations from the recent draft of “Speech and Language Processing” (Jurafsky and Martin, 2000). An entire week is devoted to topic modeling, and BigARTM (Vorontsov et al., 2015), a tool for topic modeling developed in MIPT, one of the top Russian universities and widely used in real-life applications. Overall practical sessions are aimed at developing text processing skills and practical coding skills.

Every week comprises both a lecture and a practical session. Lectures have a “talking head” format, so slides and pre-recorded demos are presented, while practical sessions are real-time coding sessions. The instructor writes code snippets in Jupyter notebooks and explains them at the same time. Overall every week, there are 3-5 lecture videos and 2-3 practical session videos. Weeks 3, 5, 9 are extended with coding assignments.

Weeks 7 and 9 are followed by interviews. In these interviews, one of the instructors’ talks to the leading specialist in the area. Tatyana Shavrina, one of the guests interviewed, leads an R&D team in Sber, one of the leading IT companies. The second guest, Konstantin Vorontsov, is a professor from one of the top universities. The guests are asked about their current projects and interests, career paths, what keeps them inspired and motivated, and what kind of advice they can give.

The final mark is calculated according to the formula:

$$\# \text{ of accepted coding assignment} \\ + 0.7 \text{mean}(\text{quiz assignment mark})$$

Coding assignments are evaluated on the binary scale (accepted or rejected), and quiz assignments are evaluated on the 10 point scale. To earn a certificate, the student has to earn at least 4 points.

In practical sessions, we made a special effort to introduce tools developed for processing texts in Russian. The vast majority of examples, utilized in lectures, problems, attempted during practical sessions, and coding assignments, utilized datasets in Russian. The same choice was made by Pavel Braslavski, who was the first to create an NLP course in Russian in 2017 (Braslavski, 2017). We utilized datasets in English only if Russian lacks the non-commercial and freely available datasets for the same task of high quality.

Some topics are intentionally not covered in the course. We focus on written texts and do not approach the tasks of text-to-speech and speech-to-text transformations. Low-resource languages spoken in Russia are out of the scope, too. Besides, we almost left out potentially controversial topics, such as AI ethics and green AI problems. Although we briefly touch upon potential biases in pre-trained language models, we have to leave out a large body of research in the area, mainly oriented towards the English language and the US or European social problems. Besides, little has been explored in how neural models are affected by those biases and problems in Russia.

The team of instructors includes specialists from different backgrounds in computer science and theoretical linguists. Three instructors worked on lectures, two instructors taught practical sessions, and three teaching assistants prepared home assignments and conducted question-answering sessions in the course forum.

3 Syllabus

Week 1. Introduction. The first introductory lecture consists of two parts. The first part overviews the core tasks and problems in NLP, presents the main industrial applications, such as search engines, Business Intelligence tools, and conversational engines, and draws a comparison between broad-defined linguistics and NLP. To conclude this part, we touch upon recent trends, which can be grasped easily without the need to go deep into details, such as multi-modal applications (Zhou et al., 2020), cross-lingual methods (Feng et al., 2020; Conneau et al., 2020) and computational humor (Braslavski et al., 2018; West and

Horvitz, 2019). Throughout this part lecture, we try to show NLP systems’ duality: those aimed at understanding language (or speech) and those aimed at generating language (or speech). The most complex systems used for machine translation, for example, aim at both. The second part of the lecture introduces such basic concepts as bag-of-words, count-based document vector representation, $tf-idf$ weighting. Finally, we explore bigram association measures, PMI and $t-score$. We point out that these techniques can be used to conduct an exploratory analysis of a given collection of texts and prepare input for machine learning methods.

Practical session gives an overview of text preprocessing techniques and simple count-based text representation models. We emphasize how preprocessing pipelines can differ for languages such as English and Russian (for example, what is preferable, stemming or lemmatization) and give examples of Python frameworks that are designed to work with the Russian language (`pymystem3` (Segalovich), `pymorphy2` (Korobov, 2015)). We also included an intro to regular expressions because we find this knowledge instrumental both within and outside NLP tasks.

During the first weeks, most participants are highly motivated, we can afford to give them more practical material, but we still need to end up with some close-to-life clear examples. We use a simple sentiment analysis task on Twitter data to demonstrate that even the first week’s knowledge (together with understanding basic machine learning) allows participants to solve real-world problems. At the same time, we illustrate how particular steps of text preprocessing can have a crucial impact on the model’s outcome.

Week 2. Word embeddings. The lecture introduces the concepts of distributional semantics and word vector representations. We familiarize the students with early models, which utilized singular value decomposition (SVD) and move towards more advanced word embedding models, such as `word2vec` (Mikolov et al., 2013) and `fasttext` (Bojanowski et al., 2017). We briefly touch upon the hierarchical softmax and the hashing trick and draw attention to negative sampling techniques. We show ways to compute word distance, including Euclidean and cosine similarity measures.

We discuss the difference between `word2vec`

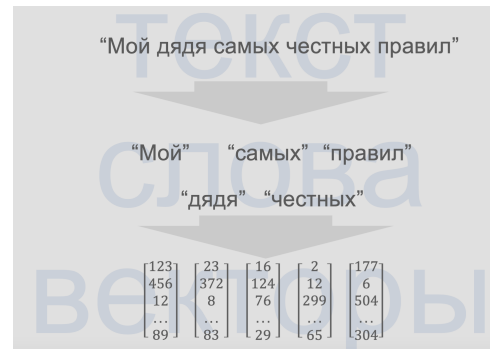


Figure 1: One slide from Lecture 2. Difference between raw texts (top line), bag-of-words (middle line), and bag-of-vectors (bottom line). Background words: text, words, vectors.

and GloVe (Pennington et al., 2014) models and emphasize main issues, such as dealing with out-of-vocabulary (OOV) words and disregarding rich morphology. `fasttext` is then claimed to address these issues. To conclude, we present approaches for intrinsic and extrinsic evaluation of word embeddings. Fig. 1 explains the difference between bag-of-words and bag-of-vectors.

In **practical session** we explore only advanced word embedding models (`word2vec`, `fasttext` and GloVe) and we cover three most common scenarios for working with such models: using pre-trained models, training models from scratch and tuning pre-trained models. Giving a few examples, we show that `fasttext` as a character-level model serves as a better word representation model for Russian and copes better with Russian rich morphology. We also demonstrate some approaches of intrinsic evaluation of models’ quality, such as solving analogy tasks (like well known “king - man + woman = queen”) and evaluating semantic similarity and some useful techniques for visualization of word embeddings space.

This topic can be fascinating for students when supplemented with illustrative examples. Exploring visualization of words clusters on plots or solving analogies is a memorable part of the “classic” NLP part of most students’ course.

Week 3. Text classification. The lecture considers core concepts for supervised learning. We begin by providing examples for text classification applications, such as sentiment classification and spam filtering. Multiple problem statements, such as binary, multi-class, and multi-label classification, are stated. To introduce ML algorithms, we start with logistic regression and move towards neu-

ral methods for text classification. To this end, we introduce `fasttext` as an easy, out-of-the-box solution. We introduce the concept of sentence (paragraph) embedding by presenting `doc2vec` model (Le and Mikolov, 2014) and show how such embeddings can be used as input to the classification model. Next, we move towards more sophisticated techniques, including convolutional models for sentence classification (Kim, 2014). We do not discuss backpropagation algorithms but refer to the DL course of the specialization to refresh understanding of neural network training. We show ways to collect annotated data on crowdsourcing platforms and speed up the process using active learning (Esuli and Sebastiani, 2009). Finally, we conclude with text augmentation techniques, including SMOTE (Chawla et al., 2002) and EDA (Wei and Zou, 2019).

In the **practical session** we continue working with the text classification on the IMDb movies reviews dataset. We demonstrate several approaches to create classification models with different word embeddings. We compare two different ways to get sentence embedding, based on any word embedding model: by averaging word vectors and using `tf-idf` weights for a linear combination of word vectors. We showcase `fasttext` tool for text classification using its built-in classification algorithm.

Additionally, we consider use `GloVe` word embedding model to build a simple Convolutional Neural Network for text classification. In this week and all of the following, we use `PyTorch`¹ as a framework for deep learning.

Week 4. Language modeling. The **lecture** focuses on the concept of language modelling. We start with early count-based models (Song and Croft, 1999) and create a link to Markov chains. We refer to the problem of OOV words and show the `add-one` smoothing method, avoiding more sophisticated techniques, such as Kneser-Ney smoothing (Kneser and Ney, 1995), for the sake of time. Next, we introduce neural language models. To this end, we first approach Bengio’s language model (Bengio et al., 2003), which utilizes fully connected layers. Second, we present recurrent neural networks and show how they can be used for language modeling. Again, we remind the students of backpropagation through time and gradient vanishing or explosion, introduced earlier in

¹<https://pytorch.org/>

the DL course. We claim, that LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Chung et al., 2014) cope with these problems. As a brief revision of the LSTM architecture is necessary, we utilize Christopher Olah’s tutorial (Olah, 2015). We pay extra attention to the inner working of the LSTM, following Andrej Karpathy’s tutorial (Karpathy, 2015). To add some research flavor to the lecture, we talk about text generation (Sutskever et al., 2011), its application, and different decoding strategies (Holtzman et al., 2019), including beam search and nucleus sampling. Lastly, we introduce the sequence labeling task (Ma and Hovy, 2016) for part-of-speech (POS) tagging and named entity recognition (NER) and show how RNN’s can be utilized as sequence models for the tasks.

The **practical session** in this week is divided into two parts. The first part is dedicated to language models for text generation. We experiment with count-based probabilistic models and RNN’s to generate dinosaur names and get familiar with perplexity calculation (the task and the data were introduced in Sequence Models course from DeepLearning.AI²). To bring things together, students are asked to make minor changes in the code and run it to answer some questions in the week’s quiz assignment.

The second part of the session demonstrates the application of RNN’s to named entity recognition. We first introduce the BIO and BIOES annotation schemes and show frameworks with pre-trained NER models for English (Spacy³) and Russian (Natasha⁴) languages. Further, we move on to CNN-biLSTM-CRF architecture described in the lecture and test it on CoNLL 2003 shared task data (Sang and De Meulder, 2003).

Week 5. Machine Translation. This **lecture** starts with referring to the common experience of using machine translation tools and a historical overview of the area. Next, the idea of encoder-decoder (seq2seq) architecture opens the technical part of the lecture. We start with RNN-based seq2seq models (Sutskever et al., 2014) and introduce the concept of attention (Bahdanau et al., 2015). We show how attention maps can be used for “black box” interpretation. Next, we reveal the core architecture of modern NLP, namely, the

²<https://www.coursera.org/learn/nlp-sequence-models>

³<https://spacy.io>

⁴<https://natasha.github.io>

Transformer model (Vaswani et al., 2017) and ask the students explicitly to take this part seriously. Following Jay Allamar’s tutorial (Alammar, 2015), we decompose the transformer architecture and go through it step by step. In the last part of the lecture, we return to machine translation and introduce quality measures, such as WER and BLEU (Papineni et al., 2002), touch upon human evaluation and the fact that BLEU correlates well with human judgments. Finally, we discuss briefly more advanced techniques, such as non-autoregressive models (Gu et al., 2017) and back translation (Hoang et al., 2018). Although we do not expect the student to comprehend these techniques immediately, we want to broaden their horizons so that they can think out of the box of supervised learning and autoregressive decoding.

In the first part of **practical session** we solve the following task: given a date in an arbitrary format transform it to the standard format “dd-mm-yyyy” (for example, “18 Feb 2018”, “18.02.2018”, “18/02/2018” → “18-02-2018”). We adopt the code from PyTorch machine translation tutorial⁵ to our task: we use the same RNN encoder, RNN decoder, and its modification - RNN encoder with attention mechanism - and compare the quality of two decoders. We also demonstrate how to visualize attention weights.

The second part is dedicated to the Transformer model and is based on the Harvard NLP tutorial (Klein et al., 2017) that decomposes the article “Attention is All You Need” (Vaswani et al., 2017). Step by step, like in the lecture, we go through the Transformer code, trying to draw parallels with a simple encoder-decoder model we have seen in the first part. We describe and comment on every layer and pay special attention to implementing the attention layer and masking and the shapes of embeddings and layers.

Week 6. Sesame Street I. The sixth **lecture** and the next one are the most intense in the course. The paradigm of pre-trained language models is introduced in these two weeks. The first model to discuss in detail is ELMo (Peters et al., 2018). Next, we move to BERT (Devlin et al., 2019) and introduce the masked language modeling and next sentence prediction objectives. While presenting BERT, we briefly revise the inner working of Trans-

⁵https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html



Figure 2: To spice up the lectures, the lecturer is dressed in an ELMo costume

former blocks. We showcase three scenarios to fine-tune BERT: (i) text classification by using different pooling strategies ([CLS], max or mean), (ii) sentence pair classification for paraphrase identification and for natural language inference, (iii) named entity recognition. SQuAD-style question-answering, at which BERT is aimed too, as avoided here, as we will have another week for QA systems. Next, we move towards GPT-2 (Radford et al.) and elaborate on how high-quality text generation can be potentially harmful. To make the difference between BERT’s and GPT-2’s objective more clear, we draw parallels with the Transformer architecture for machine translation and show that BERT is an encoder-style model, while GPT-2 is a decoder-style model. We show Allen NLP (Gardner et al., 2018) demos of how GPT-2 generates texts and how attention scores implicitly resolve coreference.

In this week, we massively rely on Jay Allamar’s (Alammar, 2015) tutorial and adopt some of these brilliant illustrations. One of the main problems, though, rising in this week is the lack of Russian terminology, as the Russian-speaking community has not agreed on the proper ways to translate such terms as “contextualized encoder” or “fine-tuning”. To spice up this week, we were dressed in Sesame Street kigurumis (see Fig. 2).

The main idea of the **practical session** is to demonstrate ELMo and BERT models, considered

earlier in the lecture. The session is divided into two parts, and in both parts, we consider text classification, using ELMo and BERT models, respectively.

In the first part, we demonstrate how to use ELMo word embeddings for text classification on the IMBdb dataset used in previous sessions. We use pre-trained ELMo embeddings by AllenNLP (Gardner et al., 2018) library and implement a simple recurrent neural network with a GRU layer on top for text classification. In the end, we compare the performance of this model with the scores we got in previous sessions on the same dataset and demonstrate that using ELMo embeddings can improve model performance.

The second part of the session is focused on models based on Transformer architecture. We use huggingface-transformers library (Wolf et al., 2020) and a pre-trained BERT model to build a classification algorithm for Google play applications reviews written in English. We implement an entire pipeline of data preparation, using a pre-trained model and demonstrating how to fine-tune the downstream task model. Besides, we implement a wrapper for the BERT classification model to get the prediction on new text.

Week 7. Sesame Street II. To continue diving into the pre-trained language model paradigm, the **lecture** first questions, how to evaluate the model. We discuss some methods to interpret the BERT's inner workings, sometimes referred to as BERTology (Rogers et al., 2021). We introduce a few common ideas: BERT's lower layers account for surface features, lower to middle layers are responsible for morphology, while the upper-middle layers have better syntax representation (Conneau and Kiela, 2018). We talk about ethical issues (May et al., 2019), caused by pre-training on raw web texts. We move towards the extrinsic evaluation of pre-trained models and familiarize the students with GLUE-style evaluations (Wang et al., 2019b,a). The next part of the lecture covers different improvements of BERT-like models. We show how different design choices may affect the model's performance in different tasks and present RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019) as members of a BERT-based family. We touch upon the computational inefficiency of pre-trained models and introduce lighter models, including DistillBERT (Sanh et al., 2019). To be solid, we touch upon other techniques to compress

pre-trained models, including pruning (Sajjad et al., 2020) and quantization (Zafir et al., 2019), but do not expect the students to be able to implement these techniques immediately. We present the concept of language transferring and introduce multilingual Transformers, such as XLM-R (Conneau et al., 2020). Language transfer becomes more and more crucial for non-English applications, and thus we draw more attention to it. Finally, we cover some of the basic multi-modal models aimed at image captioning and visual question answering, such as the unified Vision-Language Pre-training (VLP) model (Zhou et al., 2020).

In the **practical session** we continue discussing BERT-based models, shown in the lectures. The session's main idea is to consider different tasks that may be solved by BERT-based models and to demonstrate different tools and approaches for solving them. So the practical session is divided into two parts. The first part is devoted to named entity recognition. We consider a pre-trained cross-lingual BERT-based NER model from the DeepPavlov library (Burtsev et al., 2018) and demonstrate how it can be used to extract named entities from Russian and English text. The second part is focused on multilingual zero-shot classification. We consider the pre-trained XLM-based model by HuggingFace, discuss the approach's key ideas, and demonstrate how the model works, classifying short texts in English, Russian, Spanish, and French.

Week 8. Syntax parsing. The **lecture** is devoted to computational approaches to syntactic parsing and is structured as follows. After a brief introduction about the matter and its possible applications (both as an auxiliary task and an independent one), we consider syntactic frameworks developed in linguistics: dependency grammar (Tesnière, 2015) and constituency grammar (Bloomfield, 1936). Then we discuss only algorithms that deal with dependency parsing (mainly because there are no constituency parsers for Russian), so we turn to graph-based (McDonald et al., 2005) and transition-based (Aho and Ullman, 1972) dependency parsers and consider their logics, structure, sorts, advantages, and drawbacks. Afterward, we familiarize students with the practical side of parsing, so we introduce syntactically annotated corpora, Universal Dependencies project (Nivre et al., 2016b) and some parsers which perform for Russian well (UDPipe (Straka and Straková, 2017),

DeepPavlov Project (Burtsev et al., 2018)). The last part of our lecture represents a brief overview of the problems which were not covered in previous parts: BERTology, some issues of web-texts parsing, latest advances in computational syntax (like enhanced dependencies (Schuster and Manning, 2016)).

The **practical session** starts with a quick overview of CoNLL-U annotation format (Nivre et al., 2016a): we show how to load, parse and visualize such data on the example from the SynTagRus corpus⁶. Next, we learn to parse data with pre-trained UDPipe models (Straka et al., 2016) and Russian-language framework Natasha. To demonstrate some practical usage of syntax parsing, we first understand how to extract subject-verb-object (SVO) triples and then design a simple template-based text summarization model.

Week 9. Topic modelling The focus of this **lecture** is topic modeling. First, we formulate the topic modeling problem and ways it can be used to cluster texts or extract topics. We explain the basic probabilistic latent semantic analysis (PLSA) model (Hofmann, 1999), that modifies early approaches, which were based on SVD (Dumais, 2004). We approach the PLSA problem using the Expectation-Minimization (EM) algorithm and introduce the basic performance metrics, such as perplexity and topic coherence.

As the PLSA problem is ill-posed, we familiarize students with regularization techniques using Additive Regularization for Topic Modeling (ARTM) model (Vorontsov and Potapenko, 2015) as an example. We describe the general EM algorithm for ARTM and some basic regularizers. Then we move towards the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) and show that the maximum a posteriori estimation for LDA is the special case of the ARTM model with a smoothing or sparsifying regularizer (see Fig. 3 for the explanation snippet). We conclude the lecture with a brief introduction to multi-modal ARTM models and show how to generalize different Bayesian topic models based on LDA. We showcase classification, word translation, and trend detection tasks as multi-modal models.

In **practical session** we consider the models discussed in the lecture in a slightly different order. First, we take a closer look at Gensim realization

⁶https://universaldependencies.org/treebanks/ru_syntagrus/index.html

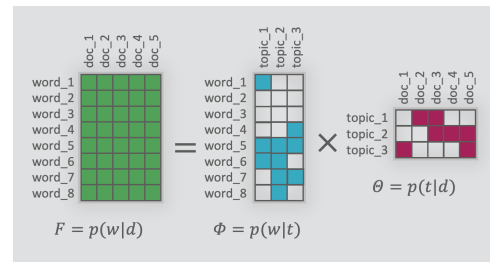


Figure 3: One slide from Lecture 9. Sparsification of an ARTM model explained.

of the LDA model (Řehůřek and Sojka, 2010), pick up the model’s optimal parameters in terms of perplexity and topic coherence, and visualize the model with pyLDAvis library. Next, we explore BigARTM (Vorontsov et al., 2015) library, particularly LDA, PLSA, and multi-modal models, and the impact of different regularizers. For all experiments, we use a corpus of Russian-language news from Lenta.ru⁷ which allows us to compare the models to each other.

Week 10. In this **lecture** we discussed monolingual seq2seq problems, text summarization and sentence simplification. We start with extractive summarization techniques. The first approach introduced is TextRank (Mihalcea and Tarau, 2004). We present each step of this approach and explain that any sentence or keyword embeddings can be used to construct a text graph, as required by the method. Thus we refer the students back to earlier lectures, where sentence embeddings were discussed. Next, we move to abstractive summarization techniques. To this end, we present performance metrics, such as ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) and briefly overview pre-Transformer architectures, including Pointer networks (See et al., 2017). Next, we show recent pre-trained Transformer-based models, which aim at multi-task learning, including summarization. To this end, we discuss pre-training approaches of T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), and how they help to improve the performance of mono-lingual seq2seq tasks. Unfortunately, when this lecture was created, multilingual versions of these models were not available, so they are left out of the scope. Finally, we talk about sentence simplification task (Coster and Kauchak, 2011; Alva-Manchego et al., 2020) and its social impact. We present SARI (Xu et al., 2016) as a

⁷<https://github.com/yutkin/Lenta.Ru-News-Dataset>

metric for sentence simplification performance and state, explain how T5 or BART can be utilized for the task.

The **practical session** is devoted to extractive summarization and TextRank algorithm. We are urged to stick to extractive summarization, as Russian lacks annotated datasets, but, at the same time, the task is demanded by in industry—extractive summarization compromises than between the need for summarization techniques and the absence of training datasets. Nevertheless, we used annotated English datasets to show how performance metrics can be used for the task. The CNN/DailyMail articles are used as an example of a dataset for the summarization task. As there is no standard benchmark for text summarization in Russian, we have to use English to measure different models' performance. We implement the TextRank algorithm and compare it with the algorithm from the NetworkX library (Hagberg et al., 2008). Also, we demonstrate how to estimate the performance of the summarization by calculating the ROUGE metric for the resulting algorithm using the PyRouge library⁸. This practical session allows us to refer back the students to sentence embedding models and showcase another application of sentence vectors.

Week 11. The penultimate **lecture** approaches Question-Answering (QA) systems and chat-bot technologies. We present multiple real-life industrial applications, where chat-bots and QA technologies are used, ranging from simple task-oriented chat-bots for food ordering to help desk or hotline automation. Next, we formulate the core problems of task-oriented chat-bots, which are intent classification and slot-filling (Liu and Lane, 2016) and revise methods, to approach them. After that, we introduce the concept of a dialog scenario graph and show how such a graph can guide users to complete their requests. Without going deep into technical details, we show how ready-made solutions, such as Google Dialogflow⁹, can be used to create task-oriented chat-bots. Next, we move towards QA models, of which we pay more attention to information retrieval-based (IR-based) approaches and SQuAD-style (Rajpurkar et al., 2016) approaches. Since natural language generation models are not mature enough (at least for Russian) to be used in free dialog, we explain

how IR-based techniques imitate a conversation with a user. Finally, we show how BERT can be used to tackle the SQuAD problem. The lecture is concluded by comparing industrial dialog assistants created by Russian companies, such as Yandex.Alisa or Mail.ru Marusya.

In the **practical session** we demonstrate several examples of using Transformer-based models for QA task. Firstly, we try to finetune Electra model (Clark et al., 2020) on COVID-19 questions dataset¹⁰ and BERT on SQuAD 2.0 (Rajpurkar et al., 2018) (we use code from huggingface tutorial¹¹ for the latter). Next, we show an example of usage of pretrained model for Russian-language data from DeepPavlov project. Finally, we explore how to use BERT for joint intent classification and slot filling task (Chen et al., 2019).

Week 12. The last **lecture** wraps up the course by discussing knowledge graphs (KG) and some of their applications for QA systems. We revise core information extraction problems, such as NER and relation detection, and show how they can be used to extract a knowledge graph from unstructured texts (Paulheim, 2017). We touch upon the entity linking problem but do not go deep into details. To propose to students an alternative view to information extraction, we present machine reading comprehension approaches for NER (Li et al., 2019a) and relation detection (Li et al., 2019b), referring to the previous lecture. Finally, we close the course by revising all topics covered. We recite the evolution of text representation models from bag-of-words to BERT. We show that all the problems discussed throughout the course fall into one of three categories: (i) text classification or sentence pair classification, (ii) sequence tagging, (iii) sequence-to-sequence transformation. We draw attention to the fact that the most recent models can tackle all of the problem categories. Last but not least we revise, how all of these problem statements are utilized in real-life applications.

The **practical session** in this week is dedicated to information extraction tasks with Stanford CoreNLP library (Manning et al., 2014). The session's main idea is to demonstrate using the tool for constructing knowledge graphs based on natural text. We consider different ways of using the

⁸[urlhttps://github.com/andersjo/pyrouge](https://github.com/andersjo/pyrouge)

⁹<https://cloud.google.com/dialogflow>

¹⁰<https://github.com/xhlulu/covid-qa>

¹¹https://huggingface.co/transformers/custom_datasets.html#question-answering-with-squad-2-0

library and experimented with using the library to solve different NLP tasks that were already considered in the course: tokenization, lemmatization, POS-tagging, and dependency parsing. The library includes models for 53 languages, so we consider examples of solving these tasks for English and Russian texts. Besides, relation extraction is considered using the Open Information Extraction (OpenIE) module from the CoreNLP library.

4 Home works

The course consists of multiple ungraded quiz assignments, 11 graded quiz assignments, three graded coding assignments. Grading is performed automatically in a Kaggle-like fashion.

4.1 Quiz Assignments

Every video lecture is followed by an ungraded quiz, consisting of 1-2 questions. A typical question address the core concepts introduced:

- What kind of vectors are more common for word embedding models?
A1: dense (true), A2: sparse (false)
- What kind of layers are essential for GPT-2 model?
A1: transformer stacks (true), A2: recurrent layers (false), A3: convolutional layers (false), A4: dense layers (false)

A graded test is conducted every week, except the very last one. It consists of 12-15 questions, which we tried to split into three parts, being more or less of the same complexity. First part questions about main concepts and ideas introduced during the week. These questions are a bit more complicated than after video ones:

- What part of an encoder-decoder model solves the language modeling problem, i.e., the next word prediction?
A1: encoder (false), A2: decoder (true)
- What are the BPE algorithm units?
A1: syllables (false), A2: morphemes (false), A3: n -grams (true), A4: words (false)

Second part of the quiz asks the students to conduct simple computations by hand:

- Given a detailed description of an neural architecture, compute the number of parameters;

- Given a gold-standard NER annotation and a system output, compute token-based and span-based micro F_1 .

The third part of the quiz asks to complete a simple programming assignment or asks about the code presented in practical sessions:

- Given a pre-trained language model, compute perplexity of a test sentence
- Does DeepPavlov cross-lingual NER model require to announce the language of the input text?

For convenience and to avoid format ambiguity, all questions are in multiple-choice format. For questions, which require a numerical answer, we provided answer options in the form of intervals, with one of the endpoints excluded.

Each quiz is estimated on a 10 point scale. All questions have equal weights.

The final week is followed by a comprehensive quiz covering all topics studied. This quiz is obligatory for those students who desire to earn a certificate.

4.2 Coding assignments

There are three coding assignments concerning the following topics: (i) text classification, (ii) sequence labeling, (iii) topic modeling. Assignments grading is binary. Text classification and sequence labeling assignments require students to beat the score of the provided baseline submission. Topic modeling assignment is evaluated differently.

All the coding tasks provide students with the starter code and sample submission bundles. The number of student's submissions is limited. Sample submission bundles illustrate the required submission format and could serve as the random baseline for each task. Submissions are evaluated using the Moodle¹² (Dougiamas and Taylor, 2003) CodeRunner¹³ (Lobb and Harlow, 2016) plugin.

4.2.1 Text classification and sequence labeling coding assignments

Text classification assignment is based on the Harry Potter and the Action Prediction Challenge from Natural Language dataset (Vilares and Gómez-Rodríguez, 2019), which uses fiction fantasy texts. Here, the task is the following: given

¹²<https://moodle.org/>

¹³<https://coderunner.org.nz/>

some text preceding a spell occurrence in the text, predict this spell name. Students are provided with starter code in Jupyter notebooks (Pérez and Granger, 2007). Starter code implements all the needed data pre-processing, shows how to implement the baseline Logistic Regression model, and provides code needed to generate the submission.

Students' goal is to build three different models performing better than the baseline. The first one should differ from the baseline model by only hyperparameter values. The second one should be a Gradient Boosting model. The third model to build is a CNN model. All the three models' predictions on the provided testing dataset should be then submitted to the scoring system. Submissions, where all the models beat the baseline models classification F1-score, are graded positively.

Sequence labeling Sequence labeling assignment is based on the LitBank data (Bamman et al., 2019). Here, the task is to given fiction texts, perform a NER labeling. Students are provided with a starter code for data pre-processing and submission packaging. Starter code also illustrates building a recurrent neural model using the PyTorch framework, showing how to compose a single-layer unidirectional RNN model.

Students' goal is to build a bidirectional LSTM model that would outperform the baseline. Submissions are based on the held-out testing subset provided by the course team.

4.2.2 Topic modeling assignment

Topic modeling assignment motivation is to give students practical experience with LDA (Blei et al., 2003) algorithm. The assignment is organized as follows: first, students have to download and pre-process Wikipedia texts.

Then, the following experiment should be conducted. The experiment consists of training and exploring an LDA model for the given collection of texts. The task is to build several LDA models for the given data: models differ only in the configured number of topics. Students are asked to explore the obtained models using the pyLDAvis (Sievert and Shirley, 2014) tool. This stage is not evaluated. Finally, students are asked to submit the topic labels that LDA models assign to words provided by the course team. Such a prediction should be performed for each of the obtained models.

5 Platform description

The course is hosted on OpenEdu¹⁴ - an educational platform created by the Association "National Platform for Open Education", established by leading Russian universities. Our course and all courses on the platform are available free of charge so that everyone can access all materials (including videos, practical Jupyter notebooks, tests, and coding assessments). The platform also provides a forum where course participants can ask questions or discuss the material with each other and lecturers.

6 Expected outcomes

First of all, we expect the students to understand basic formulations of the NLP tasks, such as text classification, sentence pair modeling, sequence tagging, and sequence-to-sequence transformation. We expect the students to be able to recall core terminology and use it fluently. In some weeks, we provide links to extra materials, mainly in English, so that the students can learn more about the topic themselves. We hope that after completing the course, the students become able to read those materials. Secondly, we anticipate that after completing the course, the students are comfortable using popular Python tools to process texts in Russian and English and utilize pre-trained models. Thirdly, we hope that the students can state and approach their tasks related to NLP, using the knowledge acquired, conducting experiments, and evaluating the results correctly.

7 Feedback

The early feedback we have received so far is positive. Although the course has only been advertised so far to a broader audience, we know that there are two groups interested in the course. First, some students come to study at their own will. Secondly, selected topics were used in offline courses in an inverse classroom format or as additional materials. The students note that our course is a good starting point for studying NLP and helps navigate a broad range of topics and learn the terminology. Some of the students note that it was easy for them to learn in Russian, and now, as they feel more comfortable with the core concepts, they can turn to read detailed and more recent sources. Unfortunately, programming assignments turn out to be our weak

¹⁴<https://npoed.ru/>

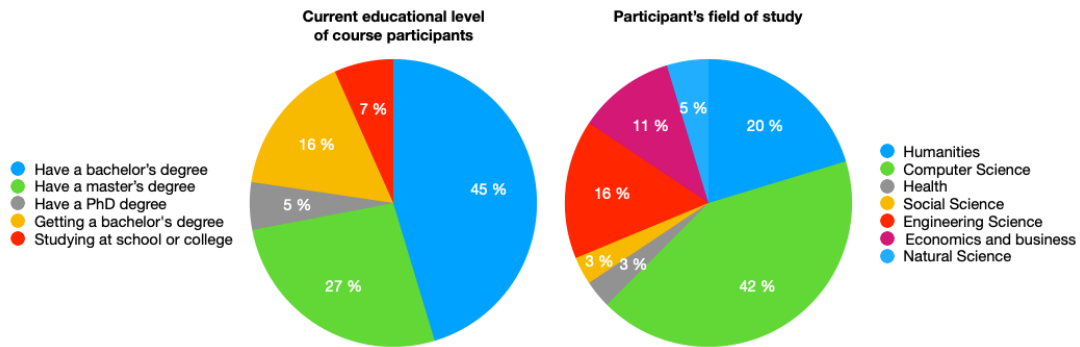


Figure 4: The results of survey among course participants. Left: current educational level. Right: professional area.

spot, as there are challenging to complete, and little feedback on them can be provided.

We ask all participants to fill in a short survey after they enroll in the course. So far, we have received about 100 responses. According to the results, most students (78%) have previously taken online courses, but only 24% of them have experience with courses from foreign universities. The average age of course participants is 32 years; most of them already have or are getting a higher education (see Fig. 4 for more details). Almost half of the students are occupied in Computer Science area, 20% have a background in Humanities, followed by Engineering Science (16%).

We also ask students about their motivation in the form of a multiple-choice question: almost half of them (46%) stated that they want to improve their qualification either to improve at their current job (33%) or to change their occupation (13%), and 20% answered they enrolled the course for research and academic purposes. For the vast majority of the student, the reputation of HSE university was the key factor to select this course among other available.

8 Conclusion

This paper introduced and described a new massive open online course on Natural Language Processing targeted at Russian-speaking students. This twelve-week course was designed and recorded during 2020 and launched by the end of the year. In the lectures and practical session, we managed to document a paradigm shift caused by the discovery and widespread use of pre-trained Transformer-based language models. We inherited the best of two worlds, showing how to utilize both static word embeddings in a more traditional machine learning setup and contextualized word embeddings in the

most recent fashion. The course's theoretical outcome is understanding and knowing core concepts and problem formulations, while the practical outcome covers knowing how to use tools to process text in Russian and English.

Early feedback we got from the students is positive. As every week was devoted to a new topic, they did not find it difficult to keep being engaged. The ways we introduce the core problem formulations and showcase different tools to process texts in Russian earned approval. What is more, the presented course is used now as supplementary material in a few off-line educational programs to the best of our knowledge.

Further improvements and adjustments, which could be made for the course, include new home works related to machine translation or monolingual sequence-to-sequence tasks and the development of additional materials in written form to support mathematical calculations, avoided in the video lecture for the sake of time.

References

- Alfred V Aho and Jeffrey D Ullman. 1972. The theory of parsing, translation, and compiling.
- Jay Alammar. 2015. The illustrated transformer. *Jay Alammar blog*.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.
- Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

- David Bamman, Sejal Popat, and Sheng Shen. 2019. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2138–2144.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.
- Leonard Bloomfield. 1936. Language or ideas? *Language*, pages 89–95.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Pavel Braslavski. 2017. *Nlp – how will it be in russian? Habr blog*.
- Pavel Braslavski, Vladislav Blinov, Valeria Bolotova, and Katya Pertsova. 2018. How to evaluate humorous response generation, seriously? In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, pages 225–228.
- Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yurii Kuratov, Denis Kuznetsov, et al. 2018. Deeppavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127.
- Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.
- Qian Chen, Zhu Zhuo, and Wen Wang. 2019. [BERT for joint intent classification and slot filling](#). *CoRR*, abs/1902.10909.
- Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: Pre-training text encoders as discriminators rather than generators](#). In *ICLR*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- William Coster and David Kauchak. 2011. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Martin Dougiamas and Peter Taylor. 2003. Moodle: Using learning communities to create an open source course management system.
- Susan T Dumais. 2004. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230.
- Andrea Esuli and Fabrizio Sebastiani. 2009. Active learning strategies for multi-label text classification. In *European Conference on Information Retrieval*, pages 102–113. Springer.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.
- Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

- Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
- Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- T HOFMANN. 1999. Probabilistic latent semantic analysis. In *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI), 1999*, pages 289–296.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.
- Daniel Jurafsky and James H Martin. 2000. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.
- Andrej Karpathy. 2015. The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*, 21:23.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). *CoRR*, abs/1408.5882.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. [Opennmt: Open-source toolkit for neural machine translation](#). In *Proc. ACL*.
- Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing*, volume 1, pages 181–184. IEEE.
- Mikhail Korobov. 2015. Morphological analyzer and generator for russian and ukrainian languages. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 320–332. Springer.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019a. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.
- Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019b. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Richard Lobb and Jenny Harlow. 2016. Coderunner: A tool for assessing computer programming skills. *ACM Inroads*, 7(1):47–51.
- Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.
- Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 91–98.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016a. **Universal Dependencies v1: A multilingual treebank collection**. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016b. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.
- Christopher Olah. 2015. Understanding lstm networks. *Christopher Olah blog*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Fernando Pérez and Brian E. Granger. 2007. **IPython: a system for interactive scientific computing**. *Computing in Science and Engineering*, 9(3):21–29.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. **Know what you don't know: Unanswerable questions for squad**. *CoRR*, abs/1806.03822.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. <http://is.muni.cz/publication/884893/en>.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor man's bert: Smaller and faster transformer models. *arXiv preprint arXiv:2004.03844*.
- Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Sebastian Schuster and Christopher D Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Ilya Segalovich. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine.
- Carson Sievert and Kenneth Shirley. 2014. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70.
- Fei Song and W Bruce Croft. 1999. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321.

- Milan Straka, Jan Hajič, and Jana Straková. 2016. [UD-Pipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).
- Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipeline. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99.
- Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *ICML*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27:3104–3112.
- Lucien Tesnière. 2015. *Elements of Structural Syntax*. John Benjamins Publishing Company.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.
- David Vilares and Carlos Gómez-Rodríguez. 2019. [Harry Potter and the action prediction challenge from natural language](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2124–2130, Minneapolis, Minnesota. Association for Computational Linguistics.
- Konstantin Vorontsov, Oleksandr Frei, Murat Apishev, Peter Romov, and Marina Dudarenko. 2015. Bigartm: Open source library for regularized multi-modal topic modeling of large collections. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 370–381. Springer.
- Konstantin Vorontsov and Anna Potapenko. 2015. Additive regularization of topic models. *Machine Learning*, 101(1):303–323.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.
- Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389.
- Robert West and Eric Horvitz. 2019. Reverse-engineering satire, or “paper on computational humor accepted despite making serious advances”. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7265–7272.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.
- Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.
- Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified Vision-language Pre-training for Image Captioning and VQA. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13041–13049.