Federal state autonomous educational institution for higher education
«Moscow institute of physics and technology
(national research university)»

On the rights of a manuscript

Le The Anh

DEEP NEURAL NETWORK MODELS FOR SEQUENCE LABELING AND COREFERENCE TASKS

Specialty 05.13.01 - «System analysis, control theory, and information processing (information and technical systems)»

Synopsis
submitted in requirements for the degree of
candidate of technical sciences

Supervisor:
PhD of physical and mathematical sciences
Burtsev Mikhail Sergeevich

Dolgoprudny - 2020

The dissertation was approved at the Cognitive dynamic systems laboratory - Moscow institute of physics and technology.

Scientific supervisor: PhD of physical and mathematical sciences,
Burtsev Mikhail Sergeevich

Leading organization: Federal research center
"Computer science and control" of Russian academy of sciences

The defence of the dissertation will be held on November 18, 2020 at 10:00 at the meeting of the dissertation council ФРКТ.05.13.01.010 at the Moscow institute of physics and technology, located at 9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russian Federation.

The dissertation can be found in the library and on the website of Moscow institute of physics and technology: `https://mipt.ru/education/post-graduate/soiskateli-tekhnicheskie-nauki.php`

The work was submitted on August 10, 2020 to the Attestation committee of the Federal State Autonomous Educational Institution of Higher Education “Moscow institute of physics and technology (national research university)” for consideration by the dissertation committee for the candidate of science and doctor of science degrees in accordance with the paragraph 3.1 of article 4 of the federal law “On science and state policy in science and technology”.

Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)»

На правах рукописи

Ле Тхе Ань

ГЛУБОКИЕ НЕЙРОСЕТЕВЫЕ МОДЕЛИ ДЛЯ ЗАДАЧ РАЗМЕТКИ ПОСЛЕДОВАТЕЛЬНОСТИ И РАЗРЕШЕНИЯ КОРЕФЕРЕНЦИИ

Специальность 05.13.01 – «Системный анализ, управление и обработка информации (информационные и технические системы)»

Автореферат

диссертации на соискание учёной степени
кандидата технических наук

Научный руководитель:

кандидат физико-математических наук
Бурцев Михаил Сергеевич

Долгопрудный - 2020

Работа прошла апробацию в лаборатории когнитивных динамических систем - Московский физико-технический институт (национальный исследовательский университет)

Научный руководитель: кандидат физико-математических наук
Бурцев Михаил Сергеевич

Ведущая организация: Федеральный исследовательский центр
"Информатика и управление"Российской академии наук

Защита состоится «18» ноября 2020 г. в 10:00 на заседании диссертационного совета ФРКТ.05.13.01.010 по адресу 141701, Московская область, г. Долгопрудный, Институтский переулок, д. 9.

С диссертацией можно ознакомиться в библиотеке и на сайте Московского физико-технического института (национального исследовательского университета): `https://mipt.ru/education/post-graduate/soiskateli-tekhnicheskie-nauki.php`

Работа представлена «10» августа 2020 г. в Аттестационную комиссию федерального государственного автономного образовательного учреждения высшего образования «Московский физико-технический институт (национальный исследовательский университет)» для рассмотрения советом по защите диссертаций на соискание ученой степени кандидата наук, доктора наук в соответствии с п.3.1 ст. 4 Федерального закона «О науке и государственной научно-технической политике».

1 Introduction

1.1 Scientific actuality of the research

Deep neural network models have recently received tremendous attention from both academy and industry, and of course, garnered amazing results in a variety of domains ranging from Computer Vision, Speech Recognition to Natural Language Processing (NLP). They significantly lifted the performance of machine learning-based systems to a whole new level, close to the human-level performance. As a matter of course, the number of deep learning projects has also increased year by year. The IPavlov project[1], based at the Neural Networks and Deep Learning Lab of Moscow Institute of Physics and Technology (MIPT), is one of them, aiming at building a set of pre-trained network models, predefined dialogue system components and pipeline templates. This dissertation is based on the work carried out as a part of this project, focusing on studying deep neural network models to address Sequence Labeling and Coreference Resolution tasks.

Sequence Labeling task is a kind of pattern recognition task involving assignment a categorical label to each element in an observed sequence. This task plays an important role in the field of Natural Language Processing (NLP) since many NLP tasks are related to labeling sequence data. Typical tasks include Named Entity Recognition (NER), Part Of Speech (POS), Speech Recognition, as well as Word Segmentation. Therefore, solving the Sequence Labeling task has been attracting a lot of attention from NLP researchers.

Coreference Resolution is an NLP task aiming at finding all mentions that refer to the same entity in a text. This task has an important role in a lot of higher-level NLP tasks including Question Answering, Text Summarization, and Information Extraction as well. However, it is one of the hardest tasks since the accurate prediction requires the model to deeply understand the meaning of the whole input text that could be a couple of sentences or even a document with hundreds of sentences. Based one that, mentions could be determined and clustered. This is a grand challenge. So far, not many works have been published and the achieved results are still not impressive compared with the other NLP tasks. Hence, there still remains a large potential for better solutions.

1.2 The goal and task of the dissertation

The main goal of this dissertation is to study supervised learning models, focusing on deep neural network models such as Bidirectional Long Short-Term Memory (Bi-LSTM), Bidirectional Gated Recurrent Unit (Bi-GRU), Convolutional Neural Network (CNN), Attention-based models, as well as state-of-the-art language models, to address Sequence Labeling and Coreference Resolution tasks. The major tasks in this PhD work include:

[1] `https://ipavlov.ai/`

- Systematically review publications on application of neural network to sequence labeling and coreference resolution tasks with focus on recent supervised deep neural networks;
- Propose and implement new models for sequence labeling task with focus on extracting useful features for the word vector representation;
- Propose and implement new models for coreference resolution focusing on dealing with the long-term dependency problem;
- Apply the proposed models to three specific tasks: Named Entity Recognition, Sentence Boundary Detection, Coreference Resolution task;
- Based on the conducted experiments on given tasks, analyse performance of the proposed models.

1.3 Related works

For Sequence Labeling task, before the advent of Deep Learning, two common approaches are rule-based and feature-based approaches. Rule-based approach, as its name indicates, is made up of the hand-crafted rules which are based on syntactic features, grammatical features, as well as domain-specific gazetteers. In the rule-based models, the rules are manually designed and then directly used for making decisions. The feature-based approach is an extended variant of the rule-based approach in which the rules are designed to represent training examples as feature vectors which are vectors of boolean or numeric values. This step is called feature engineering. The features could be character-level features, word-level features, or even document features depending on each specific task demands. Once feature vectors are built, machine learning algorithms such as Hidden Markov Model, Maximum Entropy, or Conditional Random Field are used to train a model being able to work well on unseen data. Both rule-based and feature-based approaches have the same disadvantage of time-consuming and costly rule design process. Thanks to the deep learning approach, this limitation is resolved. Deep learning-based models trained on large corpora can learn complex features due to non-linear activation functions. Moreover, one important advantage of deep learning-based models is the powerful ability of end-to-end representation learning that eliminates the need for hand-crafted rule design and feature engineering as well as simplifies sequence processing pipeline. In general, deep learning-based sequence labeling models share similar architecture with three components including (1) input encoder, (2) context encoder, and (3) tag decoder. However, the implementations of these components and the ways they are combined are very diverse. Table 1 shows some recent sequence labeling models for NER task and their obtained results on the CoNLL-2003 dataset. Generally, these

Year	Author	Model	F1
2015	Zhiheng Huang et al.	Bi-LSTM-CRF + Senna emb. + Gazetteer	90.10
2016	Onur Kuru et al.	5-layer Bi-LSTM on character-level	84.52
2016	Guillaume Lample et al.	Char. Bi-LSTM + Word Bi-LSTM	90.94
2016	Zhilin Yang et al.	Char. Bi-GRU + Word Bi-GRU + Gazetteer Joint Training (POS, Chunk, NER)	91.20
2017	Emma Strubell et al.	Iterated Dilated CNN	90.54
2018	Minghao Wu et al.	Char. CNN + Word Bi-LSTM + CRF AutoEncoder for hand-crafted feature reconstruction	91.89
2018	Peters et al.	ELMo	92.22
2018	Devlin et al	BERT	92.80
2018	Akbik et al.	Flair	93.09

Table 1: NER approaches and obtained results on CoNLL-2003 dataset.

models utilize: (1) pre-trained word embedding models (GloVe, for example) to initialize word vector representations, (2) recurrent neural networks (Bi-directional Long Short-Term Memory (Bi-LSTM, Bi-GRU) to extract context information, (3) CNN to generate a word vector representation from its characters.

For Coreference Resolution task, there are two common approaches including rule-based approach and deep learning-based approach. The rule-based approach focuses on manually building a set of rules that aids in finding mentions, searching and eliminating antecedents, and pairing them as well. Building rules that capture both syntactic and semantic features often requires a lot of labor and knowledge of language experts. Therefore, implementing a rule-based model is time-consuming and costly, and in some cases is not feasible. Moreover, such models are hard to adapt to other languages. Fortunately, learning-based models are able to overcome these long-standing shortcomings due to the ability to automatically learn rules from training data. Some recent coreference resolution models along with their obtained performance on the OntoNotes dataset are shown in table 2. These models use different algorithms to detect and cluster mentions such as Mention-Pair, Mention-Ranking, Mention-Ranking with global features learned by RNN, Mention-Ranking using end-to-end deep neural networks.

1.4 Scientific novelty

In this dissertation, we introduce a hybrid model for sequence labeling tasks which is different from previous approaches in:

- Generating rich semantic and syntactic word embeddings by combining (1) pre-trained word embedding, (2) character-level word embedding using a deep CNN network, and

Year	Author	Model	F1
2015	Olga Uryupina et al.	Mention-Pair model	61.82
2015	Wiseman et al.	Mention-Ranking model	63.39
2016	Wiseman et al.	Mention-Ranking model + Global Features (learned by RNN)	64.20
2017	Kenton Lee et al.	End-to-end First-Order model	67.20
2018	Kenton Lee et al.	Higher-Order model with Coarse-to-fine antecedent pruning	72.90
2019	Fei et al.	Deep Reinforcement Learning-based model	73.80

Table 2: Coreference Resolution approaches and obtained results on OntoNotes dataset.

(3) additional features such as POS and Chunk;

- Encoding capitalization features of whole input sequence with Bi-LSTM;
- Improving the word embedding quality by integrating and finetuning context-based word embedding.

Application of the proposed model on NER task achieves state of the art performance on Russian and Vietnamese datasets (99.17%, 94.43% on NE3 and VLSP-2016 datasets). SBD task can be reformulated as a sequence labeling task and well handled by the proposed model (89.99%, 95.88% F1 on the Cornell Movie-Dialog and DailyDialog datasets). In addition, in this dissertation, we propose a new approach to coreference resolution task which is different from previous approaches in:

- Enhancing mention detection by utilizing modern language models;
- Improving mention detection and mention clustering by learning sentence-level coreferential relations.

Application of the model with sentence coreference module for Russian language achieve state of the art of 58.42% average F1 on RuCor dataset.

1.5 Theoretical and practical value of the work in the dissertation

- Proposed sequence labeling models including WCC-NN-CRF and ELMo WCC-NN-CRF can be applied to several NLP tasks such as Named Entity Recognition, Part of Speech, Chunking, and Sentence Boundary Detection as well.
- The WCC-NN-CRF and ELMo WCC-NN-CRF models are implemented in the open-source DeepPavlov framework[2]. Trained NER models on Vietnamese and English

[2] `https://deeppavlov.ai`

datasets are available to download at `https://github.com/deepmipt/DeepPavlov/tree/master/deeppavlov/configs/ner`. These models are already to use or can be further fine-tuned on specific domains.

- The SBD model trained on conversational datasets can be used in the annotating phase of a socialbot. The trained SBD model on the DailyDialog dataset can be found at `https://github.com/deepmipt/DeepPavlov/tree/master/deeppavlov/configs/sentence_segmentation`. This trained model was integrated into our socialbot which was built to participate in the Alexa Prize - Socialbot Grand Challenge 3.

- The Sentence-level Coreferential Relation-based model can be used to extract the sentence relationship in a document-level context and has a promising potential not only for coreference resolution task but also for question answering, relation extraction or any other task that needs the information about the relationship between sentences.

1.6 Statements to be defended

- WCC-NN-CRF model for sequence labeling task which utilizes (1) a CNN to generate vector representation of words from their characters and (2) a Bi-LSTM to encode the capitalization features of an input sequence. This model achieved state of the art performance on Vietnamese and Russian datasets with 98.21%, 94.43% on NE3 and VLSP-2016;

- Extensions of WCC-NN-CRF with a context-based word vector representation generated by modern language models. This model obtained cutting edge performance on Russian and English datasets with 99.17%, 92.91%, and 92.27% F1 on NE3, Gareev's, and CoNLL-2003 datasets;

- Sentence Boundary Detection task can be reformulated as a sequence labeling task and addressed by proposed sequence labeling models with impressive results of 89.99%, 95.88% on the Cornell Movie-Dialog and DailyDialog datasets;

- Sentence-level Coreferential Relation-based (SCRb) model to extract the sentence relationship in the coreference context;

- Coreference resolution models extended with SCRb obtained state of the art of 58.42% average F1 on RuCor dataset;

1.7 Presentations and validation of the research results

The main findings and contributions of the dissertation were presented and discussed at four conferences:

- Artificial Intelligence and Natural Language Conference, ITMO University, Saint Petersburg, Russia, September 20-23, 2017.
- The 2nd Asia Conference on Machine Learning and Computing, Ton Duc Thang University, Ho Chi Minh, Viet Nam, December 7 - 9, 2018.
- The 25th International Conference on Computational Linguistics and Intellectual Technologies, Russian State University for Humanities, Moscow, Russian, May 29 - June 01, 2019.
- The 4th International Conference on Machine Learning and Soft Computing, Mercure, Haiphong, Vietnam, January 17-19, 2020.

1.8 Publications

The proposed models applied to NER, SBD, and Coreference Resolution tasks along with the achieved results were published in five papers, out of which the first four papers were already indexed by SCOPUS:

1. **Le T.A.**, Arkhipov M. Y., Burtsev M. S., "Application of a Hybrid Bi- LSTM- CRF Model to the Task of Russian Named Entity Recognition". In: Filchenkov A., Pivovarova L., Zizka J. (eds) Artificial Intelligence and Natural Language. AINL 2017. Communications in Computer and Information Science, pp. 91–103. 2017. Url: `https://link.springer.com/chapter/10.1007/978-3-319-71746-3_8`
2. **Le T. A.** and Burtsev M. S., "A Deep Neural Network Model for the Task of Named Entity Recognition". In: International Journal of Machine Learning and Computing. Vol. 9. 1., pp. 8–13. 2019. Url: `http://www.ijmlc.org/vol9/758-ML0025.pdf`
3. **Le T. A.**, Kuratov Y. M. , Petrov M. A., Burtsev M. S., "Sentence Level Representation and Language Models in the task of Coreference Resolution for Russian". In: 25th International Conference on Computational Linguistics and Intellectual Technologies. 2019. Url: `http://www.dialog-21.ru/media/4609/letaplusetal-160.pdf`
4. **Le T. A.**, "Sequence Labeling Approach to the Task of Sentence Boundary Detection". In: Proceedings of the 4th International Conference on Machine Learning and Soft Computing. New York, NY, USA: Association for Computing Machinery, pp. 144–148. 2020. Url: `https://dl.acm.org/doi/10.1145/3380688.3380703`

5. Yuri Kuratov, Idris Yusupov, Dilyara Baymurzina, Denis Kuznetsov, Daniil Cherniavskii, Alexander Dmitrievskiy, Elena Ermakova, Fedor Ignatov, Dmitry Karpov, Daniel Kornev, **The Anh Le**, Pavel Pugin, Mikhail Burtsev, "DREAM technical report for the Alexa Prize 2019". In: The 3rd Proceedings of Alexa Prize. 2020. Url: `https://m.media-amazon.com/images/G/01/mobile-apps/dex/alexa/alexaprize/assets/challenge3/proceedings/Moscow-DREAM.pdf`

Personal contribution of the author in the works with co-authors is as follows:

1. Development and implementation of a hybrid Bi-LSTM CRF model for solving the task of Named Entity Recognition in Russian.

2. Development and implementation of a character-aware WCC-NN-CRF model and its application to the task of Named Entity Recognition for four languages: Vietnamese, Russian, English, and Chinese.

3. Development and implementation of the Sentence-level Coreferential Relation-based (SCRb) model for determining the relationship of sentences in the context of coreference; integration of the SCRb model into the baseline coreference model for solving the task of Coreference Resolution for Russian language.

5. Implementation and integration of the ELMo WCC-NN-CRF model for NER and SBD tasks in the socialbot architecture.

2 The content of the dissertation

The full dissertation is presented in 141 pages and organized into five chapters:

- Chapter 1 is an introductory chapter that firstly gives a brief overview of Deep Learning and Natural Language Processing, and then briefly summaries the dissertation structure and contributions;

- Chapter 2 focuses on representing some major concepts of Deep Learning knowledge that are closely related to the proposed models described in the next two sections. The first one is the word embedding models that are responsible for converting raw words into vectors which can be processed by neural networks. The second concept is about deep neural network models which are mostly used for NLP tasks. The last one is general-purpose language models that aim at building language models which can be applied to most of NLP task to boost the model performance. Applying these models is easy and requires a little effort;

- Chapter 3 describes proposed models for the task of Sequence Labeling, and conducted experiments on NER and SBD tasks;

- Chapter 4 introduces a new approach to the task of Coreference Resolution, which aims at building a model to extract the sentence-level coreferential relationship. This model can be used in two ways: (1) output of this model is used as an additional feature for the baseline model, (2) this model is jointly trained with the baseline model;
- Chapter 5 summarizes the completed works, achieved results, and conclusions as well.

Below, only two main chapters, the third and fourth chapters, are presented.

2.1 Sequence labeling with character-aware deep neural networks and language models

The goal of the Sequence Labeling task is to determine an appropriate state for each observation. From the supervised machine learning perspective, the Sequence Labeling task can be formally defined as follow. Given the training data:

$$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\},$$

where:

$$x^{(i)} = x_1 \ldots x_n, x_j \in \mathbf{X},$$

$$y^{(i)} = y_1 \ldots y_n, y_j \in \mathbf{Y},$$

here $\mathbf{X}$ and $\mathbf{Y}$ denote the sets of all possible observations and states. The task, here, is to build a predictor by learning a map function $f : x \rightarrow y$ that works well on unseen input sequence x.

2.1.1 Backbone WCC-NN-CRF Architecture

The family of models proposed and studied in this dissertation has the same core backbone architecture. This architecture follows a hybrid approach and consists of three encoding sub-networks to (1) encode the semantic and grammatical features of words, (2) capture character-level word representation as well as capitalization features of words, and (3) capture the meaning of words in their contexts. In addition, a CRF layer is utilized to exploit the output tag dependencies. A graphical illustration of the proposed model is shown in Fig. 1. We refer to this backbone model as WCC-NN-CRF in the text.

Let $V_{word}, V_{char}, V_{cap}$ be word, character, and capitalization type vocabularies, respectively; here $|V|$ denotes the size of the vocabulary V. In the proposed model, three kinds of lookup tables are used to map words, characters, and capitalization types of words to dense vectors. Let denote them as $\mathbf{L}_{word} \in \mathbb{R}^{|V_{word}| \times d_{word}}$, $\mathbf{L}_{char} \in \mathbb{R}^{|V_{char}| \times d_{char}}$, $\mathbf{L}_{cap} \in \mathbb{R}^{|V_{cap}| \times d_{cap}}$. Here d_{word}, d_{char}, and d_{cap} are the lengths of dense vectors representing words, character, and capitalization type of word, respectively. $\mathbf{L}_{word}$ is initialized by a pre-trained word embedding (Glove, for example). $\mathbf{L}_{char}$ and $\mathbf{L}_{cap}$ are initialized randomly with values drawn from

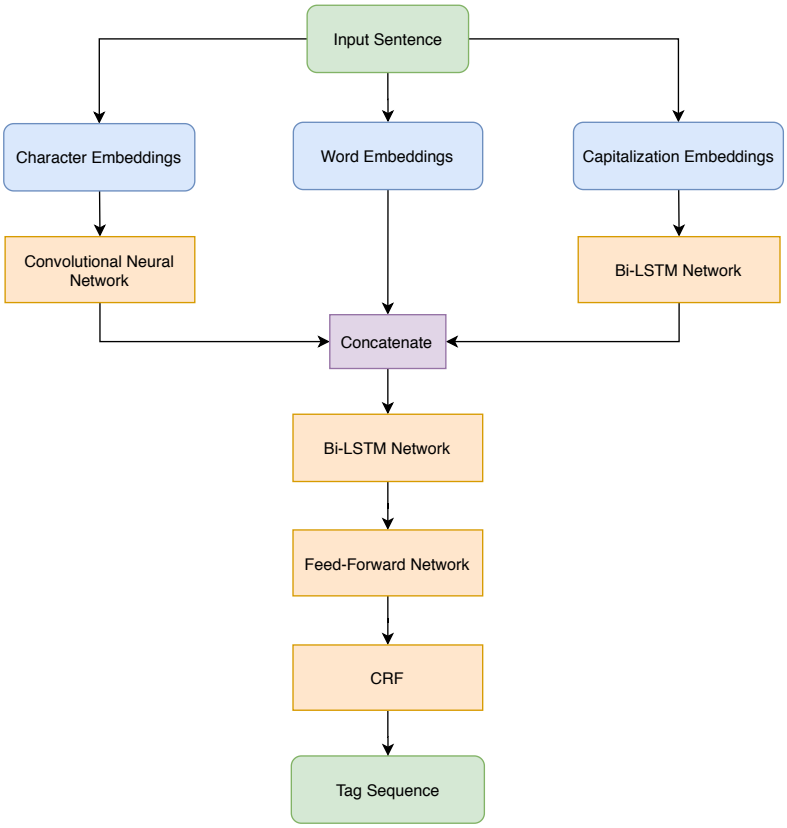

Figure 1: Proposed WCC-NN-CRF model for Sequence Labeling task.

a uniform distribution with range [-0.5, 0.5]. All of these lookup tables are then fine-tuned during training state.

Given an input sentence with n words $\mathbf{x} = \{x_1, .., x_n\}$, it is is transformed into the word, character, and capitalization index forms: $\mathbf{x}_{word} \in \mathbb{R}^n, \mathbf{x}_{char} \in \mathbb{R}^{n \times nb_{char}}$, $\mathbf{x}_{cap} \in \mathbb{R}^n$. Here, nb_{char} denotes the number of characters in a word. The padding technique is used to ensure that all words have the same number of characters. After that, word, capitalization and character embeddings $\mathbf{e}_{word} \in \mathbb{R}^{n \times d_{word}}, \mathbf{e}_{cap} \in \mathbb{R}^{n \times d_{cap}}$, $\mathbf{e}_{char} \in \mathbb{R}^{n \times nb_{char} \times d_{char}}$ are created by looking up $\mathbf{x}_{word}$, $\mathbf{x}_{cap}$, $\mathbf{x}_{char}$ in $\mathbf{L}_{word}$, $\mathbf{L}_{cap}$, and $\mathbf{L}_{char}$, respectively.

The mapping from character and capitalization embeddings to word embeddings are detailed below.

Character feature extraction with CNN. Recall that $\mathbf{e}_{char} \in \mathbb{R}^{n \times nb_{char} \times d_{char}}$ are vector embeddings representing characters of the input sentence. These vectors are fed into three convolutional layers with different window sizes [1, 3], [1, 4], and [1, 5]. Let denote them as: $\mathbf{f}_1 \in \mathbb{R}^{3 \times d_{char} \times d_{conv_1}}$, $\mathbf{f}_2 \in \mathbb{R}^{4 \times d_{char} \times d_{conv_2}}$, $\mathbf{f}_3 \in \mathbb{R}^{5 \times d_{char} \times d_{conv_3}}$. Let $\mathbf{O}_1 \in \mathbb{R}^{n \times nb_{char} \times d_{conv_1}}$, $\mathbf{O}_2 \in \mathbb{R}^{n \times nb_{char} \times d_{conv_2}}$, and $\mathbf{O}_3 \in \mathbb{R}^{n \times nb_{char} \times d_{conv_3}}$ denote the outputs of these convolutional layers. Each position (i, j) on the t^{th} slice of these outputs are computed using the equation 1. The proposed model uses (1, 1) slide along with padding technique to ensure that the outputs of convolutional layers have the same shape with the inputs. In other words, the number of words and number of characters remain unchanged after applying convolutional

layers.

$$\mathbf{O}[i,j,t] = \sum_{k=0}^{d_{char}-1} \sum_{r=r_{left}}^{r_{right}} \mathbf{e}[i,j+r,k] \times \mathbf{f}[r+\lfloor f_w/2 \rfloor, k, t] + \mathbf{b}[k,t], \tag{1}$$

where:

$$r_{left} = -\lfloor f_w/2 \rfloor, \tag{2}$$

$$r_{right} = \lfloor f_w/2 \rfloor - (f_w - 1) \; mod \; 2, \tag{3}$$

here $\lfloor \; \rfloor$ and *mod* denote floor division and modulo operations, respectively. $\mathbf{b}$, f_w denote biases and the filter width, respectively.

$\mathbf{O}_1, \mathbf{O}_2, \mathbf{O}_3$ are then concatenated along the features dimension to create the feature tensor $\mathbf{O} \in \mathbb{R}^{n \times nb_{char} \times d_{sum}}$. Finally, the character-based word representations are computed by reducing the character dimension:

$$\mathbf{O} = [\mathbf{O}_1, \mathbf{O}_2, \mathbf{O}_3], \tag{4}$$

$$\mathbf{e}^{char}_{word} = \text{max\_pooling}(\mathbf{O}), \tag{5}$$

here [ ] denote the concatenation operation.

Fig. 2 shows the complete illustration of the CNN network for character feature extraction.

Capitalization feature extraction with Bi-LSTM. A Bi-LSTM network is used to capture the capitalization feature of words in their left and right contexts (See Fig. 3). Input sentences are transformed into the capitalization form that encodes each word with one integer value representing the capitalization of that word. For example, the sentence: "MIPT is located in Dolgoprudny." will be encoded to "0 1 1 1 2". Here 0 value denotes the word with all characters uppercase, values of 1 mean that all characters of the word are in lowercase, and value of 2 is used for the words starting with a capitalized character. In our implementation, we use four capitalization types including: upper_case, lower_case, fist_cap, and otherwise.

Recall from the beginning of section 2.1.1 that $\mathbf{e}_{cap} \in \mathbb{R}^{n \times d_{char}}$ are capitalization vector representations of the input sentence. These vectors are fed into two Bi-LSTM networks to capture the sentence-level context of words and produce the capitalization-based vector representation of words:

$$\mathbf{e}^{cap}_{word} = [\overrightarrow{\texttt{LSTM}}(\mathbf{e}_{cap}), \overleftarrow{\texttt{LSTM}}(\mathbf{e}_{cap})], \tag{6}$$

here $\overrightarrow{\texttt{LSTM}}(\mathbf{e}_{cap})$, $\overleftarrow{\texttt{LSTM}}(\mathbf{e}_{cap})$ denote the outputs from the forward and backward LSTM layers, respectively; [ ] is the concatenation operation.

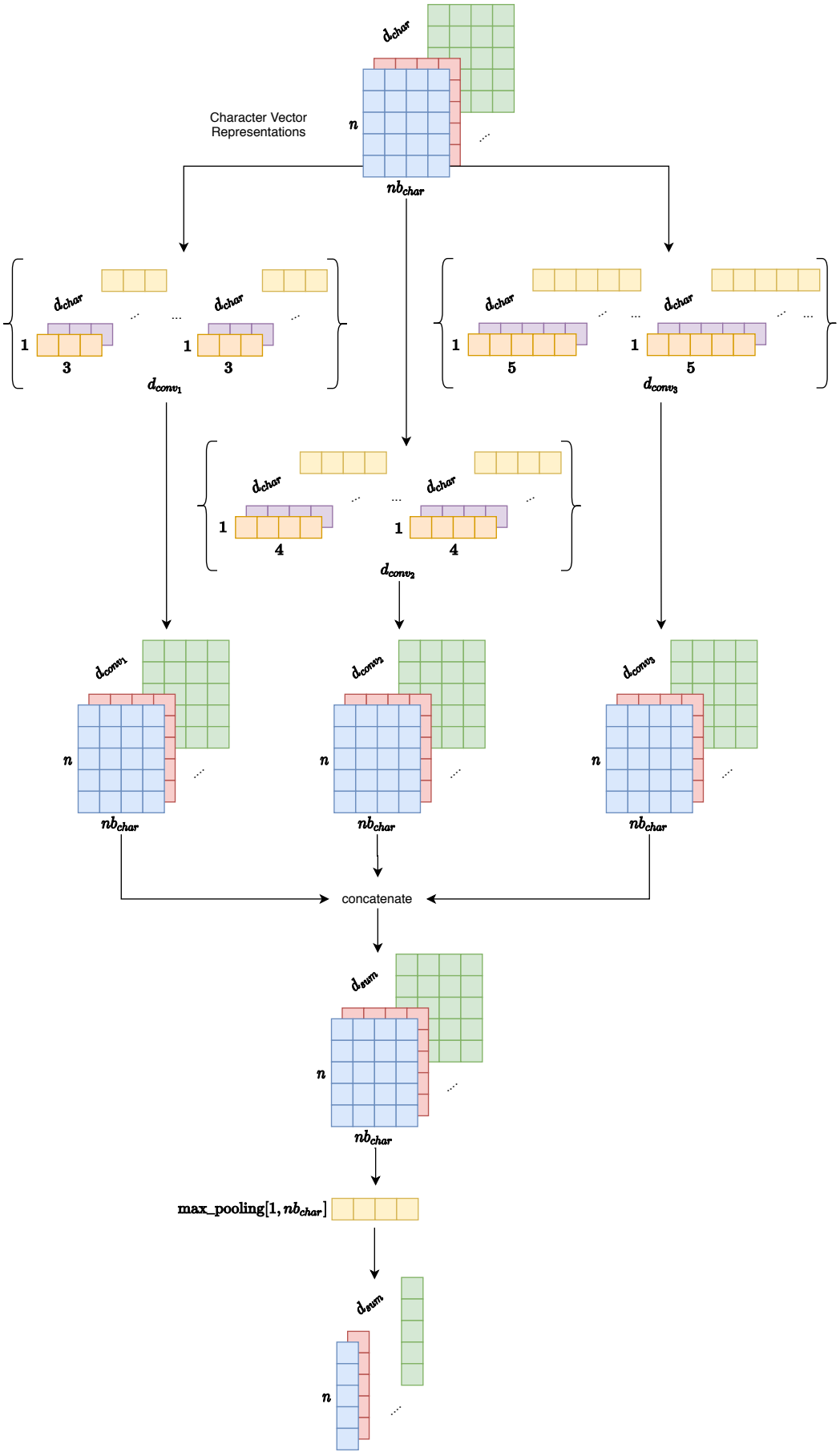

Figure 2: Character features extraction with CNN.

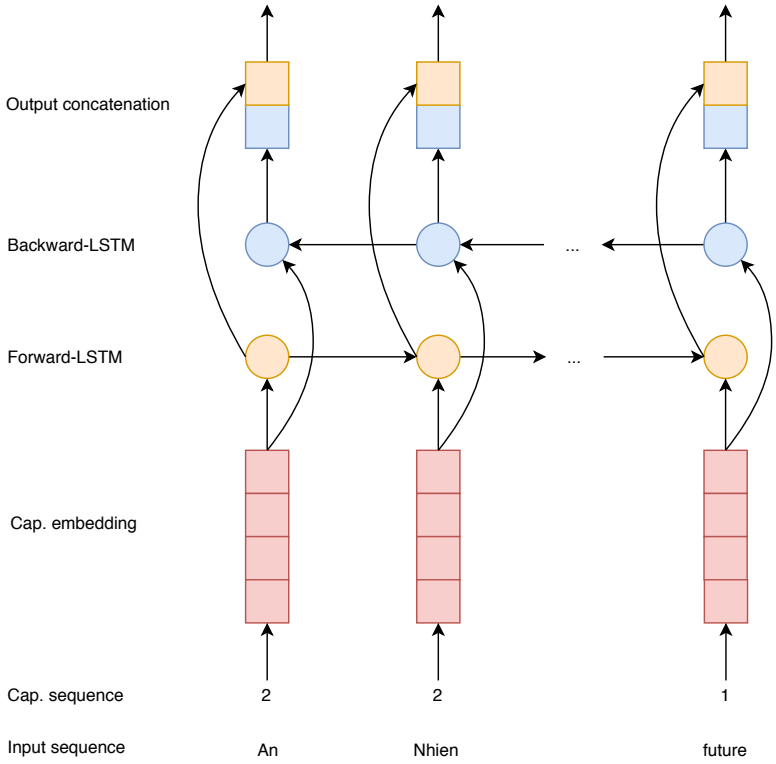

Figure 3: Extraction Capitalization Feature Extraction using Bi-LSTM Network.

Once three kinds of word vector representations $\mathbf{e}_{word}, \mathbf{e}^{char}_{word}, \mathbf{e}^{cap}_{word}$ are computed, the word vector representation is just their concatenation:

$$\mathbf{e} = [\mathbf{e}_{word}, \mathbf{e}^{char}_{word}, \mathbf{e}^{cap}_{word}]. \tag{7}$$

$\mathbf{e}$ is then fed into another Bi-LSTM network to produce the final word representation in the sentence context:

$$\mathbf{r}_{word} = [\overrightarrow{\texttt{LSTM}}(\mathbf{e}), \overleftarrow{\texttt{LSTM}}(\mathbf{e})]. \tag{8}$$

Let nb_{tags} be a number of tags predefined for the given task. A feed-forward neural network is used to produce scores of words in the input sentence according to tags $\mathbf{S}_{ffnn}[n, nb_{tags}]$. Here $\mathbf{S}_{ffnn}[i, j]$ represents the score of j^{th} tag for the i^{th} word.

Tag dependencies extraction with CRF. The word representation is a concatenation of a pre-trained word embedding, character-level word embedding produced by CNN network, and capitalization embedding generated by the Bi-LSTM network. This word representation is then fed into the second Bi-LSTM network to produce the final word representation in the left and right contexts. From now on, we have two ways to produce the final tag sequence. The first way is to directly feed the final word representation into a softmax layer to compute probabilities of tags. In this case, predicting the output tag just depends on the input sequence, but not on the previously assigned tags. The second way is to use CRF or

another recurrent neural network to make tagging decisions based on the previous and next output tags. From our experiments, CRF for extracting tag dependencies performs better than a recurrent neural network.

In addition to the $\mathbf{S}_{ffnn}$, the CRF model uses another type of score called transition score $\mathbf{T}$ that represents how likely one tag follows another. The score for the pair of the input sentence $\mathbf{x}$ and a tagging sequence $\mathbf{y} = \{y_1, .., y_n\}$ is calculated by equation below:

$$\mathbf{S}_{crf}(\mathbf{x}, \mathbf{y}) = \mathbf{T}[\mathbf{y}_0, \mathbf{y}_1] + \sum_{i=1}^{n}(\mathbf{S}_{ffnn}[i, \mathbf{y}_i] + \mathbf{T}[\mathbf{y}_i, \mathbf{y}_{i+1}]), \tag{9}$$

where $\mathbf{y}_0$, $\mathbf{y}_{n+1}$ are dummy tags added to represent the beginning and the end of the tag sequence. Hence, the transition matrix has shape of $[nb_{tags} + 2, nb_{tags} + 2]$. This transition matrix is fine tuned during a training stage.

After that, the softmax function is applied to estimate conditional probabilities of the tag sequence:

$$p(\mathbf{y}|\mathbf{x}) = \frac{e^{\mathbf{S}_{crf}(\mathbf{x},\mathbf{y})}}{\sum_{\hat{\mathbf{y}}\in\mathbf{Y}_\mathbf{x}} e^{\mathbf{S}_{crf}(\mathbf{x},\hat{\mathbf{y}})}}, \tag{10}$$

here $\mathbf{Y}_\mathbf{x}$ denotes the set of all possible output tag sequences.

During the training stage, the log-probability of the correct tag sequence is maximized. At the inference stage, the output tag sequence is the sequence that maximizes the score given by:

$$\mathbf{y}^* = argmax_{\hat{\mathbf{y}}\in\mathbf{Y}_\mathbf{x}} S_{crf}(\mathbf{x}, \hat{\mathbf{y}}). \tag{11}$$

2.1.2 Language model-based architecture

To further improve the system's performance, state-of-the-art language models such as BERT, ELMo are leveraged to enhance the quality of input word vector representations. These language models can be integrated into the model from Section 2.1.1 in two ways: (1) word vector representations are extracted from the language model and used as an additional input features, (2) language model is directly integrated into the architecture and fine-tuned during the training stage. According to our conducted experiments, treating the language model as a part of the overall architecture to fine-tune it during training for the NER task is better than just using it as an additional feature. This section presents two language model-based architectures for the NER task: (1) WCC-NN-CRF + ELMo to address the monolingual NER task, and (2) WCC-NN-CRF + BERT to deal with the multilingual task.

WCC-NN-CRF with ELMo. This model is an extension of the model described in the section 2.1.1, in which two word embedding types are used: (1) free-context word embedding like Glove, and (2) context-based word embedding like ELMo.

WCC-NN-CRF with BERT-based Multilingual Model. This section describes another proposed model for multilingual tasks. The model is a combination of the WCC-NN-CRF model described in section 2.1.1 with BERT. A BERT model[3] pre-trained on a large text corpus covered more than 100 languages is fine-tuned to produce word vector representations specific for the NER task. Besides that, a character CNN network is still used to generate word vector representations from their characters to better represent out-of-vocabulary words. The final word vector representations are the combination of BERT output and character-level word vectors. A Bi-LSTM network is then utilized to produce the word vectors in the sentence context. Finally, a feed-forward network followed by a CRF layer is used to estimate probabilities of the tags.

2.1.3 Application of WCC-NN-CRF models for Named Entity Recognition

NER is a subtask of information extraction that locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations. NER is often one of the first steps in an NLP pipeline, and plays an important role in many NLP tasks, including but not limited to Question Answering, Coreference Resolution, Topic Modeling, and Text Summarization.

In this section, we present the experiments of our WCC-NN-CRF models on NER task. To have a comprehensive analysis of WCC-NN-CRF models, six datasets were selected. These datasets cover four languages including Vietnamese, Russian, English, and Chinese: CoNLL-2003, VLSP-2016, NE3, NE5, Gareev's, and MSRA.

Table 3 shows the tagging performance of WCC-NN-CRF model on six datasets. We also compare WCC-NN-CRF model with recent Vietnamese models in table 4. Figure 4 visualizes the effectiveness of each component in WCC-NN-CRF model over three given datasets. According to the experiment results, character features extracted by the CNN network plays an important role in enhancing the model performance: about 12%, 5%, and 15% for VLSP-2016, CoNLL-2003, and Gareev's datasets, respectively. Adding a Bi-LSTM network to extract the capitalization features of words yielded further improvement, about 3%. It is interesting to find out from these experiments that using POS and Chunk features helps to significantly boost the model performance, about 5%, when testing on the VLSP-2016 dataset, whereas the improvement is almost negligible on CoNLL-2003 dataset. This partially shows that the syntactic features in the Vietnamese language play a more important role than in English in the context of the NER task.

The evaluation of ELMo WCC-NN-CRF model on CoNLL-2003, NE3, and Gareev's datasets compared with some English and Russian NER models is shown in tables 5, 6. Tagging performance of Multilingual BERT-based WCC-NN-CRF model in comparison with

[3] `https://github.com/google-research/bert`

Language	Dataset	P	R	F1
Russian	Gareev's	87.07	90.40	88.69
	NE3	98.09	98.34	98.21
	NE5	94.33	95.29	94.81
English	CoNLL-2003 *	90.91	91.52	91.22
Vietnamese	VLSP-2016 *	94.91	93.96	94.43
Chinese	MSRA	91.99	93.92	92.95

Table 3: Tagging performance of WCC-NN-CRF. * denotes using POS and Chunk features.

Model	P	R	F1
Le et al. (2016)	89.56	89.75	89.66
Pham et al. (2017)	91.09	93.03	92.05
Pham et al. (2017)	92.76	93.07	92.91
Nguyen et al. (2018)	93.87	**93.99**	93.93
WCC-NN-CRF	**94.91**	93.96	**94.43**

Table 4: Tagging performance of WCC-NN-CRF on VLSP-2016 dataset compared with some Vietnamese models.

WCC-NN-CRF and ELMo WCC-NN-CRF is shown in table 7.

2.1.4 Application of WCC-NN-CRF models for Sentence Boundary Detection

Sentence Boundary Detection (SBD) is an NLP problem of deciding where sentences begin and end in an unpunctuated text. This task has an important role in voice-enable chatbot since some automatic speech recognition devices just output unpunctuated text while many NLP tasks work at sentence-level. We firstly reformulate SBD as a sequence labeling task and then apply our ELMo WCC-NN-CRF model.

Let x be the unpunctuated text consisting of n tokens outputted by an automatic speech recognition device:

$$x = \{x_1, x_2, .., x_n\}. \tag{12}$$

Let t be the list of k predefined punctuation mark types:

$$t = \{t_1, t_2, .., t_k\}, \tag{13}$$

corresponding to the k types of sentences. The goal of SBD task is to split x into m sentences s, by adding punctuation marks:

$$s = \{s_1, s_2, .., s_m\}, \tag{14}$$

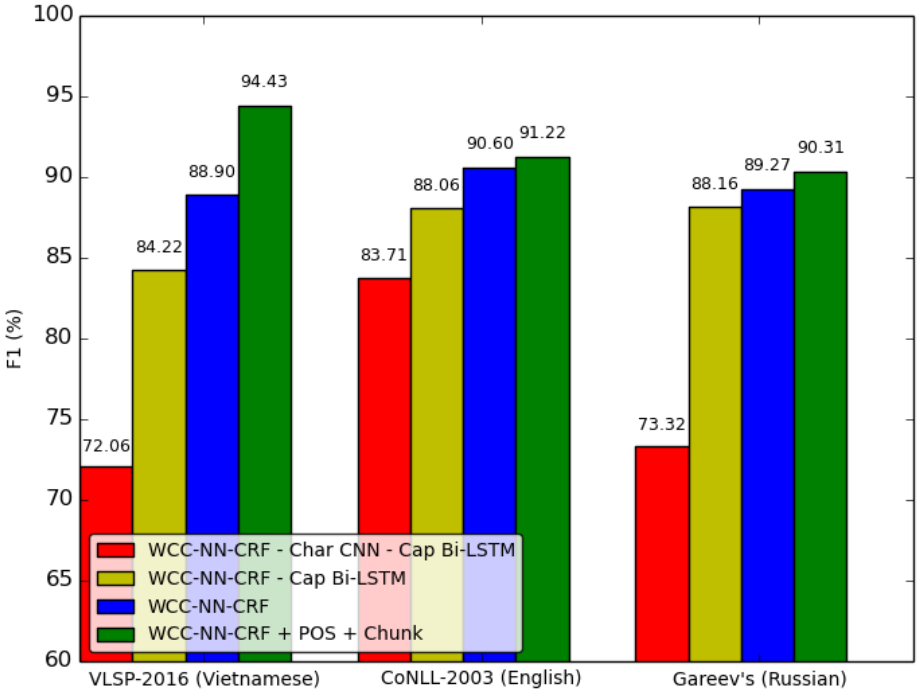

Figure 4: Importance of different components for performance of backbone WCC-NN-CRF model across the datasets.

Model	P	R	F_1
Huang et al. (2015)	-	-	90.10
Strubell et al. (2017)	-	-	90.54
Xu et al. (2017)	**92.13**	89.61	90.85
Passos et al. (2014)	-	-	90.90
Lample et al. (2016)	-	-	90.94
Luo et al. (2015)	91.50	91.40	91.20
Ma et al. (2016)	-	-	91.21
Wang et al. (2017)	91.39	91.09	91.24
Devlin et al. (2018)	-	-	92.80
Akbik et al. (2018)	-	-	**93.09**
ELMo WCC-NN-CRF (ours)	92.04	**92.50**	92.27

Table 5: Tagging performance of ELMo WCC-NN-CRF on CoNLL-2003 dataset compared with some state-of-the-art models.

Model	Gareev's			NE3			
	PER	ORG	OVERALL	PER	ORG	LOC	OVERALL
Gareev et al. (Knowledge-based model)	79.30	55.48	62.17	-	-	-	-
Gareev et al. (CRF-based model)	84.84	71.31	75.05	-	-	-	-
Malykh et al. (Character-based LSTM)	92.89	69.14	62.49	-	-	-	-
Mozharova et al. (CRF-based model)	-	-	-	96.08	83.84	94.57	91.71
Mozharova et al. (Feature-based model)	-	-	-	97.21	95.21	85.60	92.92
Romanov et al. (FastText-CNN-CRF)	-	-	-	-	-	-	95.00
Romanov et al. (BERT-based model)	-	-	-	-	-	-	96.00
Our model (WCC-NN-CRF)	96.08	85.48	88.76	99.02	97.49	97.87	98.21
Our model (ELMo WCC-NN-CRF)	**98.70**	**90.32**	**92.91**	**99.90**	**99.05**	**99.30**	**99.17**

Table 6: F1-score of ELMo WCC-NN-CRF model on Russian datasets compared with some published models.

Model	VLSP-2016 (Vietnamese)			CoNLL-2003 (English)			NE3 (Russian)		
	P	R	F_1	P	R	F_1	P	R	F_1
WCC-NN-CRF	90.61	87.25	88.90	90.44	90.76	90.60	98.08	98.34	98.21
ELMo WCC-NN-CRF	-	-	-	**92.04**	**92.50**	**92.27**	**99.05**	**99.30**	**99.17**
BERT-Multi WCC-NN-CRF	**91.24**	**90.34**	**90.79**	89.27	91.91	90.47	99.05	99.24	99.15

Table 7: Tagging performance of Multilingual BERT-based WCC-NN-CRF model on VLSP-2016, CoNLL-2003, and NE3.

where:

$$s_i = \{x_a, x_{a+1}, .., x_b, t_{s_i}\}, \tag{15}$$

$$\sum_{i=1}^{m} |s_i| = n + m, \tag{16}$$

here s_i is a segment of x after adding the corresponding punctuation mark t_{s_i}, and s_i is consecutive with the previous one in the order in x. $|s_j|$ denote the length of the punctuated sentence s_j.

We transform s into the sequence y:

$$y = \{y_1, y_2, .., y_n\}, \tag{17}$$

which has the same length with the input sequence x, by traveling through each sentence in s, each word per time, to create the corresponding tag by following below rules:

- If the current word is the first word of the sentence then replace it with the tag, t_{s_i}, corresponding to the last word in the sentence.
- Remove the last word of the sentence.
- Replace the remaining words with tag O marking that these words are not the first one.
- Finally, concatenate all modified sentences to create the tag sequence.

We use the first word in the sentence to mark the sentence boundary instead of the last one since the first word (e.g., who, what, when, how, do, am) often contains more information for determining the type of sentence.

The task now can be reformulated. Given an unpunctuated lowercase text x consisting of n tokens:

$$x = \{x_1, x_2, .., x_n\}, \tag{18}$$

the model need to predict the sequence of tags:

$$y = \{y_1, y_2, .., y_n\}, \tag{19}$$

where $y_i \in t$.

Now modification of backbone WCC-NN-CRF model can be used to address the SBD task. We experimented on two datasets generated from two conversational datasets, Cornel Movie-Dialog and DailyDialog, and achieved 89.99% and 95.88%, respectively (See table 8).

Dataset	Tag	P	R	F1
Cornell Movie-Dialog	Question	87.99	76.11	81.62
	Statement	91.27	92.54	91.90
	Overall	90.70	89.30	89.99
DailyDialog	Question	95.54	93.79	94.66
	Statement	96.14	96.43	96.29
	Overall	95.99	95.77	95.88

Table 8: Tagging performance of ELMo WCC-NN-CRF on Cornell Movie-Dialog and DailyDialog datasets.

2.2 Coreference resolution with Sentence-level Coreferential Scoring

Coreference Resolution (CR) is the task of identifying and clustering all expressions that refer to the same entity in a text. Let take two sentences below as an example:

`My sister` `has a friend called` `John` . `She` `thinks` `he` `is so funny.`

There are two people mentioned in these sentences: `My sister` and `John`. A CR model needs to find and cluster all mentions referring to these people. Hence, the output should be {`My sister, She`}, {`John, he`}. CR is very useful for information extraction, text summarization, as well as question answering systems. For example, CR plays an important role in question answering systems as it can be used to retrieve the named entities that pronouns refer to (i.e., CR enables question answering systems to answer the question: "What are we talking about?").

2.2.1 Sentence-level Coreferential Relation-based model

We propose Sentence-level Coreferential Relation-based (SCRb) model to learn the sentence relationship in the coreference context. More concretely, SCRb model takes as input a list of sentences and outputs: sentence representation $SCRb_{sr}$ and sentence-level coreferential relation score $SCRb_{ss}$ that expresses the probability of existence of coreference links between two sentences (See Fig. 5).

SCRb model use three types of word embeddings including (1) a free-context embedding e_{fc}, (2) a context-based embedding e_{cb}, and (3) a character-based embedding e_{ch} learned by a CNN network. All of these vectors are concatenated to generate the final word embedding:

$$e_w = [e_{fc}, e_{cb}, e_{ch}], \tag{20}$$

here [,] denotes the concatenation operator.

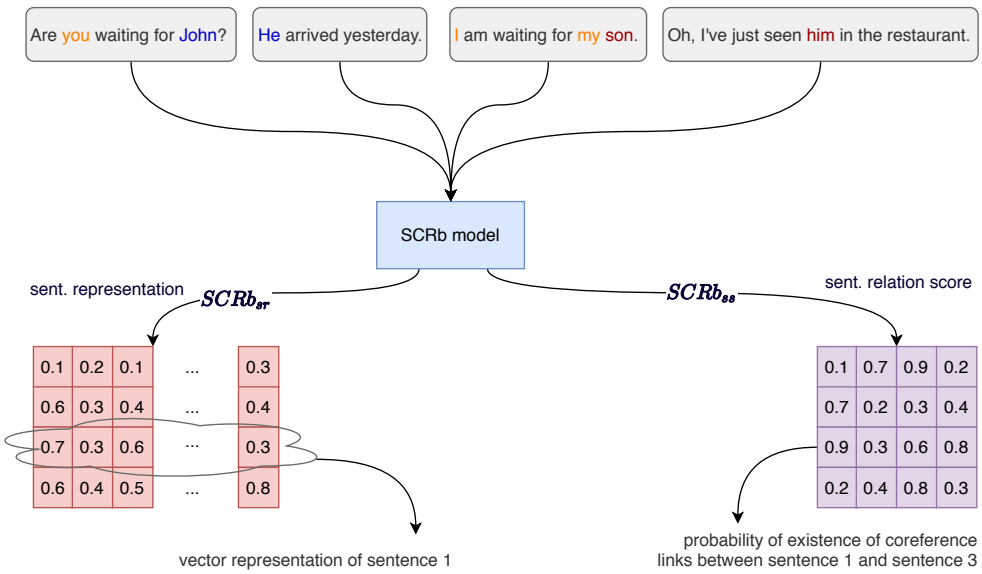

Figure 5: Sentence-level Coreferential Relation-based model

The resulting word embeddings of each sentence are then feed into a Bi-LSTM network to capture both left and right sentence contexts:

$$w = [\overrightarrow{\texttt{LSTM}}(e_w), \overleftarrow{\texttt{LSTM}}(e_w)] \tag{21}$$

here $\overrightarrow{\texttt{LSTM}}(e_w)$, $\overleftarrow{\texttt{LSTM}}(e_w)$ denote outputs of forward and backward LSTM networks, respectively.

A max pooling or attention mechanism is utilized to reduce the word dimension to generate the sentence representation:

$$s = \max\_\text{pooling}(w), \tag{22}$$

The second Bi-LSTM is then used to capture the final sentence representation in the document context:

$$SCRb_{sr} = [\overrightarrow{\texttt{LSTM}}(s), \overleftarrow{\texttt{LSTM}}(s)]. \tag{23}$$

To generate coreferential relations between sentences, we modified Multi-dimensional Self-attention proposed by Tao Shen et al. in 2018 by adding distance encoding between sentences. Let $SCRb_{sr}(i) \in \mathbb{R}^{d_s}$ be the vector representing the i^{th} sentence in the document, here d_s denotes the length of sentence vectors generated by the last Bi-LSTM network. Let $e_{d_{ij}} \in \mathbb{R}^{d_d}$ is distance embedding between sentence i and sentence j, here d_d denotes the length of position encoding vectors. Let $W \in \mathbb{R}^{d_s}$, $W_1, W_2 \in \mathbb{R}^{d_s \times d_s}$, $W_d \in \mathbb{R}^{d_s \times d_d}$ are weight matrices, and $b_1 \in \mathbb{R}^{d_s}$, $b \in \mathbb{R}$ are bias terms. The setence-level coreferential relation score between sentence i and sentence j is computed via a feed-forward neural network:

$$SCRb_{ss}(i,j) = W^T \sigma(W_1 SCRb_{sr}(i) + W_2 SCRb_{sr}(j) + W_d e_{d_{ij}} + b_1) + b, \tag{24}$$

where σ is the activation function.

The graphical illustration of SCRb model is shown in figure 6.

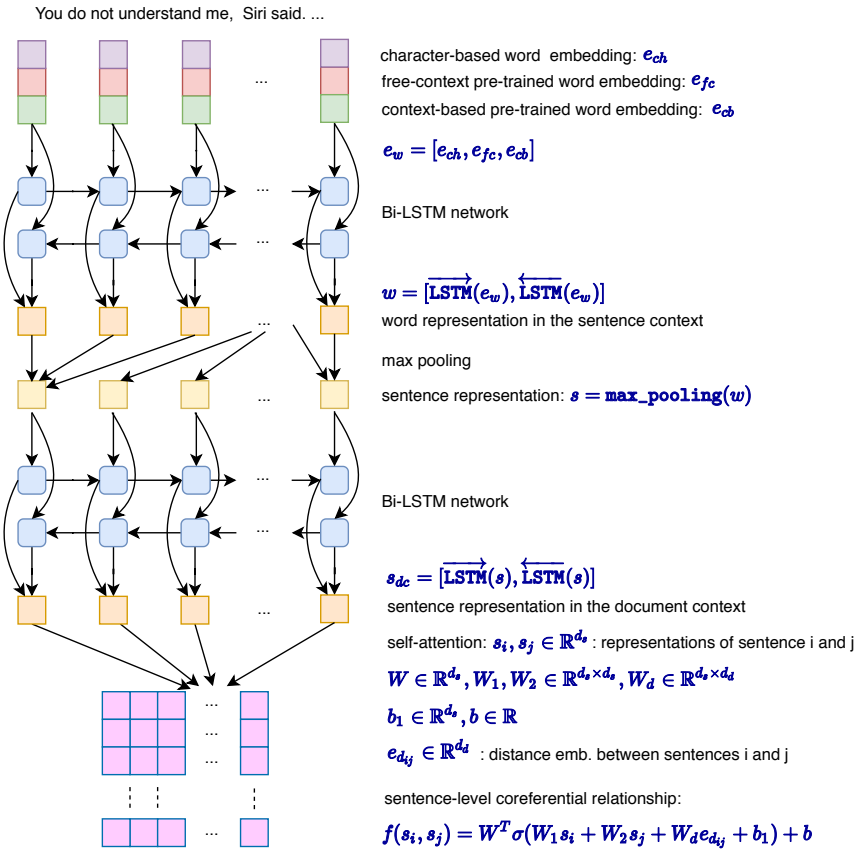

Figure 6: Sentence-level Coreferential Relation-based model

2.2.2 Proposed Coreference Resolution models

Baseline model. Among some recent CR models, we choose Kenton Lee's model as a baseline model because of its state of the art performance and code availability. In our work, we extend the baseline model and focus on building an end-to-end model for the Russian language.

The baseline model considers CR task as a set of decisions for every possible span in the document. Let D be a document of T words, $N = \frac{T(T+1)}{2}$ be the number of possible spans in D. The span i is determined by its start and end indices $\texttt{START}(i)$ and $\texttt{END}(i)$. The task is to assign each span i with an antecedent y_i. Here, $y_i \in \mathcal{Y} = \{\epsilon, 1, \ldots, i-1\}$. ϵ is a dummy antecedent representing two possible scenarios: (1) the span is not a mention or (2) the span is a mention that is not coreferent with any previous span. The goal of the model is to learn a conditional probability distribution:

$$P(y_1, \ldots, y_N | D) = \prod_{i=1}^{N} P(y_i | D) = \prod_{i=1}^{N} \frac{\exp(s(i, y_i))}{\sum_{y' \in \mathcal{Y}} \exp(s(i, y'))}, \tag{25}$$

where $s(i, j)$ is the coreferential score between span i and span j. Bellow, we step by step describe how the coreferential score is computed:

- Word vector representations, $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, are concatenation of pre-trained word em-

beddings and character-based word embeddings generated by 1-dimensional CNN networks.

- Compute contextualized word vector representation by using Bi-LSTM network:

$$\boldsymbol{x}_t^* = [\overrightarrow{\texttt{LSTM}}_t(\boldsymbol{x}), \overleftarrow{\texttt{LSTM}}_t(\boldsymbol{x})]. \tag{26}$$

- Compute vector representation of span i:

$$\boldsymbol{g}_i = [\boldsymbol{x}^*_{\texttt{START}(i)}, \boldsymbol{x}^*_{\texttt{END}(i)}, \hat{\boldsymbol{x}}_i, \phi(i)], \tag{27}$$

here $\phi(i)$ is the vector encoding the size of span i, $\hat{\boldsymbol{x}}_i$ is the weighted sum of word vectors in span i, computed by using an attention mechanism over words of span i:

$$\alpha_t = \boldsymbol{w}_\alpha \cdot \text{FFNN}_\alpha(\boldsymbol{x}_t^*), \tag{28}$$

$$\alpha_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\texttt{START}(i)}^{\texttt{END}(i)} \exp(\alpha_k)}, \tag{29}$$

$$\hat{\boldsymbol{x}}_i = \sum_{k=\texttt{START}(i)}^{\texttt{END}(i)} \alpha_{i,t} \cdot \boldsymbol{x}_t, \tag{30}$$

here FFNN denotes a feed-forward neural network.

- Use feed-forward neural networks to compute mention score, and antecedent score:

$$s_m(i) = \boldsymbol{w}_m \cdot \text{FFNN}_m(\boldsymbol{g}_i), \tag{31}$$

$$s_a(i,j) = \boldsymbol{w}_a \cdot \text{FFNN}_a([\boldsymbol{g}_i, \boldsymbol{g}_j, \boldsymbol{g}_i \circ \boldsymbol{g}_j, \phi(i,j)]), \tag{32}$$

where $\cdot$ indicates the dot product, $\circ$ denotes element-wise multiplication, and $\phi(i,j)$ is the feature vector encoding speaker and genre information from the metadata and the distance between two spans.

- Compute coreference score:

$$s(i,j) = \begin{cases} 0 & j = \epsilon \\ s_{\text{m}}(i) + s_{\text{m}}(j) + s_{\text{a}}(i,j) & j \neq \epsilon \end{cases} \tag{33}$$

The main drawback of this model lies in the pairwise scoring function $s(i,j)$ which just makes local decisions by considering only pairs of spans. To enhance this function, an iterative inference method was proposed to refine span representations. This entails refining the antecedent distributions:

$$P_n(y_i) = \frac{\exp(s(\mathbf{g}_i^n, \mathbf{g}_{y_i}^n))}{\sum_{y \in \mathcal{Y}(i)} \exp(s(\mathbf{g}_i^n, \mathbf{g}_y^n))}. \tag{34}$$

$\mathbf{g}_i^1$ is computed as described before. At the t^{th} iteration, the attention mechanism is utilized to compute the expected antecedent representation $\mathbf{a}_i^t$, and then the span representation is updated:

$$\mathbf{a}_i^t = \sum_{y_i \in \mathcal{Y}(i)} P_t(y_i) \cdot \mathbf{g}_{y_i}^t, \tag{35}$$

$$\mathbf{f}_i^t = \sigma(\mathbf{W}_f[\mathbf{g}_i^t, \mathbf{a}_i^t]), \tag{36}$$

$$\mathbf{g}_i^{t+1} = \mathbf{f}_i^t \circ \mathbf{g}_i^t + (1 - \mathbf{f}_i^t) \circ \mathbf{a}_i^t. \tag{37}$$

Baseline + SCRb model. SCRb model can be combined with the baseline model in two ways:

- Directly integrated SCRb model into the baseline model. Both models are jointly trained together;
- Use the output of SCRb model as an additional feature for the baseline model.

Figure 7 shows the way we combine the baseline model with SCRb model. The sentence representation and sentence-level coreferential relation score, $SCRb_{sr}$, $SCRb_{ss}$ outputted from SCRb model are used to improve the mention scoring and coreference scoring functions. Specifically, we modify the mention score function and coreference score function by adding $SCRb_{sr}$ and $SCRb_{ss}$:

$$s_m(i) = \boldsymbol{w}_m \cdot \text{FFNN}_m([\boldsymbol{g}_i, SCRb_{sr}(i)]), \tag{38}$$

$$s(i,j) = \alpha \times (s_m(i) + s_m(j) + s_a(i,j)) + \beta \times SCRb_{ss}(i,j), \tag{39}$$

here α, β are model hyperparameters.

Baseline + BERT model. BERT-base model has a total of 12 layers. We conducted experiments with outputs of different layers to study manually how BERT pays attention at each layer in the coreference context. Based on this analysis, we decided to take for our model outputs from three layers 1, 6, and 12 with more coreference relevant attention patterns. The word contextualized vector representations are the weighted sum of these outputs. The final word embedding is the concatenation of two types of embeddings: (1) free-context word embeddings (Glove + Character-based word embedding), and (2) the context-based word embedding generated from BERT-base model.

2.2.3 Experiments and results

Three datasets are used to evaluate two above proposed models:

- OntoNotes - an English dataset used for CoNLL-2012 shared task,
- RuCor - a Russian dataset used for Dialogue-2014 shared task,

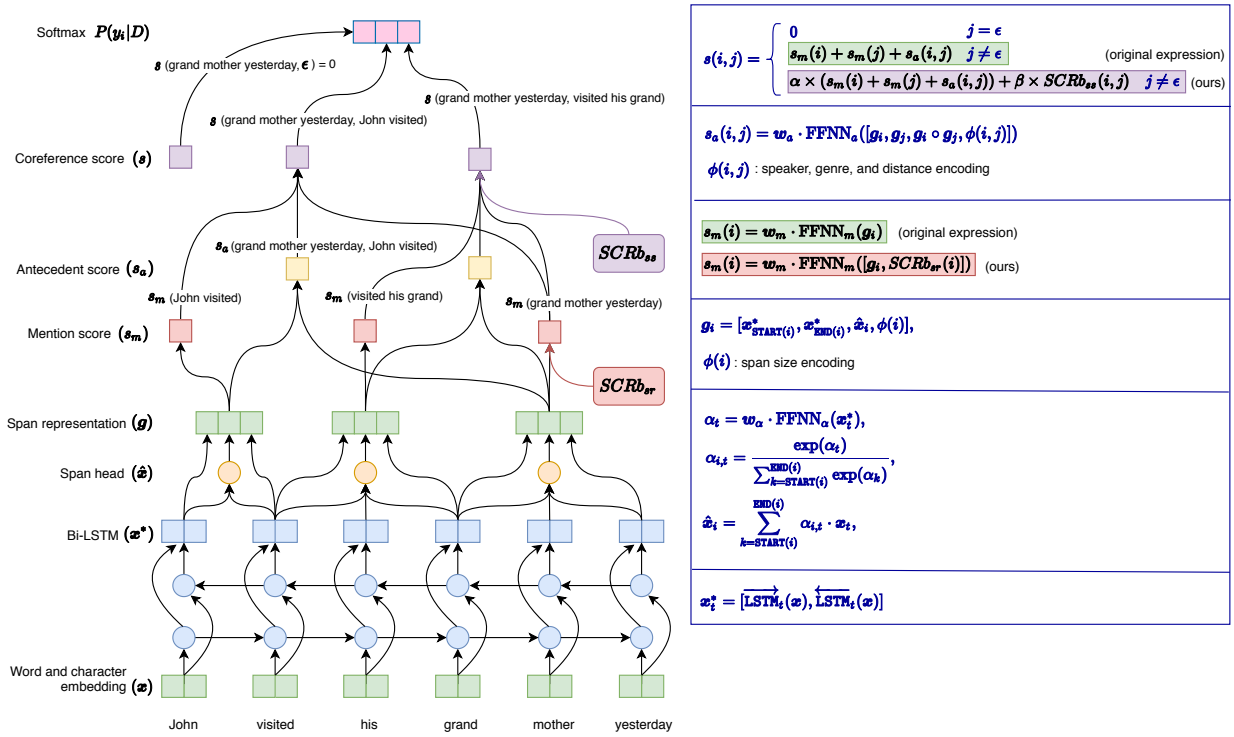

Figure 7: Baseline + SCRb model for Coreference Resolution task. Only a subset of possible spans is depicted.

Dataset	Max. F1 (%) on the dev. set
Original OntoNotes 5.0	72.90
OntoNotes 5.0 + sent.-level coref. relationship with the acc. of 80.0%	74.13
OntoNotes 5.0 + sent.-level coref. relationship with the acc. of 84.0%	74.73
OntoNotes 5.0 + sent.-level coref. relationship with the acc. of 91.0%	75.56
OntoNotes 5.0 + sent.-level coref. relationship with the acc. of 94.0%	76.36
OntoNotes 5.0 + sent.-level coref. relationship with the acc. of 96.5%	77.01
OntoNotes 5.0 + sent.-level coref. relationship with the acc. of 98.5%	77.92
OntoNotes 5.0 + ground truth sent.-level coref. relationship	78.84

Table 9: Effect of sentence-level coreferential relations on the baseline SCRb model performance.

- AnCor - a Russian dataset used for Dialogue-2019 shared task.

The first experiment aims at studying the significance of the sentence-level coreferential relations. We analyse how the information about sentence-level coreferential relation affects the baseline model performance. To do this, we train one model with exactly the same parameters on two datasets: (1) the original OntoNotes 5.0 dataset, (2) the same dataset but augmented with ground truth sentence-level coreferential relations. Different amount of noise were added in different training sessions to get better understanding of the importance of the sentence-level information. As a result we got the model's performance for a different values of accuracy of sentence relation presence. The results presented in the table 9 show that the information about the sentence relations is a very useful feature for the CR task. Specifically, if this feature is provided with an accuracy of 91%, the model performance will be increased by about 2.5%, a promising percentage. Under the ideal condition, when training with ground truth sentence-level coreferential relations, the model performance reached to 78.84%, an improvement by 5.84% compared with the performance of the baseline model trained with the original OntoNotes 5.0 dataset.

In the second experiment, we implement seven modifications of the SCRb model and test their performance on the OntoNotes 5.0 dataset:

- SCRb M1: The baseline model that is the combination of GloVe embedding, Bi-LSTM, CNN, and Self-attention.
- SCRb M2: M1 + log scale distance embedding inside self-attention.
- SCRb M3: M2 + weighted losses to deal with the unbalanced data problem.
- SCRb M4: M4 + ELMo embedding.
- SCRb M5: M3 + BERT embedding.
- SCRb M6: This is a modification of M5 by outputting non-symmetric outputs.
- SCRb M7: M5 - GloVe embedding.

The detail results of these variants on the validation set and test set are shown in Fig. 8.

Table 10 shows the evaluation of Baseline + SCRb model on OntoNotes dataset compared with some recent coreference resolution models. Testing results of our proposed model on two Russian datasets in comparison with the other Russian models are shown in table 11.

Model	Avg. F1
Martschat et al. (2015)	62.5
Clark et al. (2015)	63.0
Wiseman et al. (2015)	63.4
Wiseman et al. (2016)	64.2
Clark et al. (2016)	65.3
Clark et al. (2016)	65.7
Lee et al. (2017)	67.2
Lee et al. (2018)	72.9
Fei et al. (2019)	73.8
Baseline + SCRb M7	**74.1**

Table 10: Testing results on OntoNotes dataset.

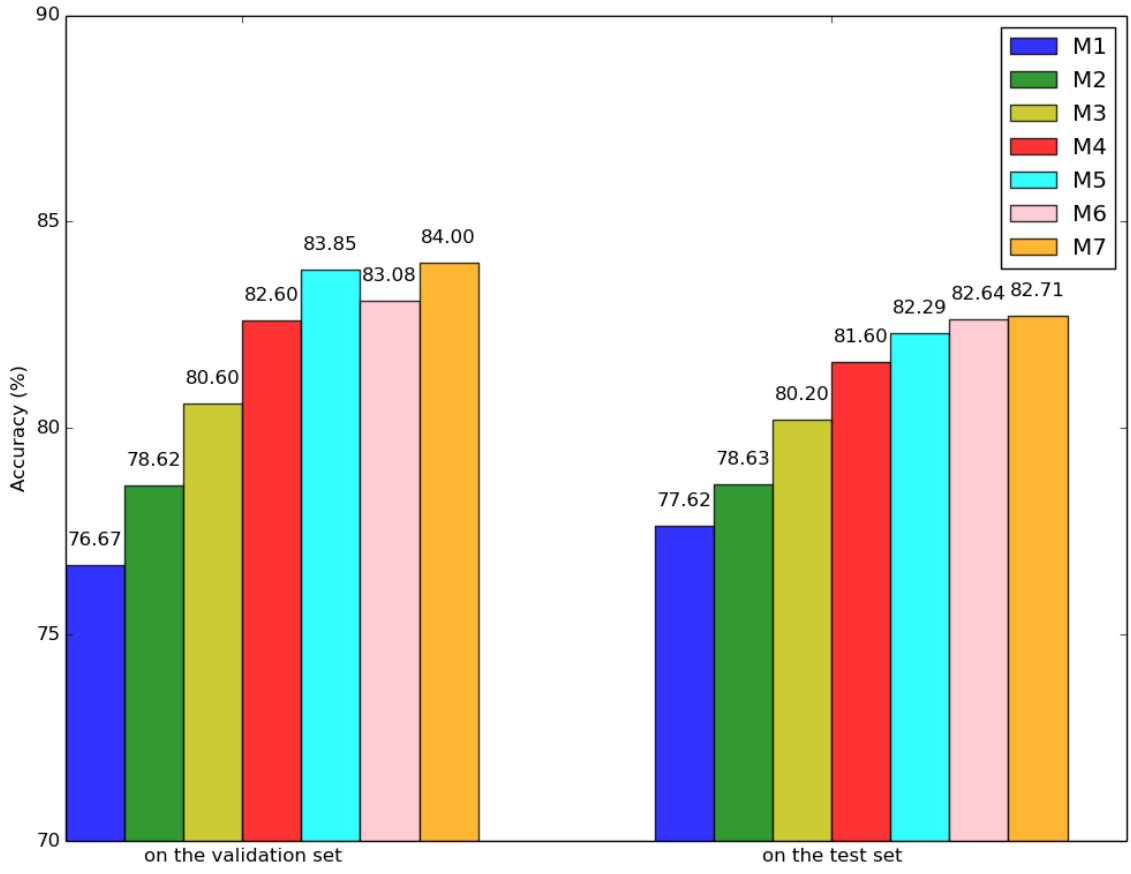

Figure 8: SCRb model's variants performance. M1: Baseline model (Bi-LSTM + CNN + Self attention). M2: M1+ log scale distance inside self attention. M3: M2 + weighted class (to deal with imbalanced data problem). M4: M3 + ELMo. M5: M4 + BERT. M6: M5 + BERT with nonsymmetric output. M7: M5 - GloVe.

Dataset	Model	Avg. F1
Rucor	Sysoev (2017)	35.09
	Baseline + ELMo*	57.58
	Baseline + SCRb	**58.42**
AnCor	Baseline + ELMo*	51.72
	Baseline + SCRb	53.61
	Baseline + BERT*	53.76
	Baseline + ELMo* + RuCor	55.96
	Baseline + BERT* + RuCor	**57.78**

Table 11: Testing results on RuCor and AnCor dataset. * denotes pre-trained models on DeepPavlov framework that are fine-tuned on Russian corpora.

3 Conclusions

In conclusion, the dissertation proposed deep learning-based models to address sequence labeling and coreference resolution tasks. Some conclusions drawn from the dissertation are listed below:

1. Current state-of-the-art approach to sequence labeling tasks is the hybrid approach that combines deep neural networks with traditional structured prediction models.

2. Word vector representation generated from the characters is a crucial feature, especially in case of small training data or the large number of out-of-vocabulary words.

3. POS and Chunk features play an important role for the NER task.

4. Fine-tuning modern language models helps to (1) improve model performance, (2) handle scarce data problem, and (3) make the model more robust and generalize.

5. Building multilingual models based on BERT is good choice for low-resource language or specific domains.

6. SBD task can be reformulated as Sequence Labeling task and well handled by ELMo WCC-NN-CRF model.

7. Sentence-level coreferential relation is a very useful feature for the coreference resolution task.

8. Position feature encoding the distance between sentences, weighted class, and modern language models are the key components in building the SCRb model that boost the model accuracy from 77% up to 84%.

9. Using modern language models such as ELMo or BERT significantly boosts the model performance and reducing the training time as well, outperform the previous models.

The main contributions of the dissertation are summarized below:

1. An original hybrid model for sequence labeling task which extends existed Bi-LSTM CRF architectures with (1) trainable CNN for generation of character-level representation of an input sequence, and (2) Bi-LSTM for encoding capitalization features; achieves SOTA performance on Russian and Vietnamese datasets with F1 98.21%, 94.43% on NE3 and VLSP-2016.

2. Extensions of the original architecture with encoders based on language models ELMo and BERT were evaluated on Russian and English datasets. It obtained state of the art performance of 92.91%, 99.17%, 92.27% F1 on Gareev's dataset, NE3, and CoNLL-2003.

3. Application of proposed sequence labeling model to the sentence boundary detection task produced solid results of 89.99% F1 and 95.88% F1 on the Cornell Movie-Dialog and DailyDialog datasets.

4. Sentence-level coreferential relation can significantly improve the performance of solving coreference resolution task. The experiments on OntoNotes dataset shows that quality of solution can be boosted up to 5.84%.

5. An original model for learning sentence-level coreferential relationships was introduced. Incorporation of this model in the baseline coreference architecture improved it's performance for English.

6. Application of the model with sentence coreference module for Russian language allowed to achieve state of the art of 58.42% average F1 on RuCor dataset.