Proceedings of Machine Learning Research 95:407-421, 2018 ACML 2018

Concorde: Morphological Agreement in Conversational Models

Daniil Polykovskiy DANIIL.POLYKOVSKIY@GMAIL.COM
Moscow State University, Moscow, Russia
National Research University Higher School of Economics, Moscow, Russia

Dmitry Soloviev D.SOLOVIEV@CORP.MAIL.RU
Mail.ru, Moscow, Russia

Sergey Nikolenko SNIKOLENKO@NEUROMATION.IO
Neuromation OU, Tallinn, Estonia
Steklov Mathematical Institute at St. Petersburg, Russia

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Neural conversational models are widely used in applications such as personal assistants and chat bots. These models seem to give better performance when operating on the word level. However, for fusional languages such as French, Russian, or Polish, the vocabulary size can become infeasible since most of the words have multiple of word forms. To reduce vocabulary size, we propose a new pipeline for building conversational models: first generate words in a standard (lemmatized) form and then transform them into a grammatically correct sentence. In this work, we focus on the *morphological agreement* part of the pipeline, i.e., reconstructing proper word forms from lemmatized sentences. For this task, we propose a neural network architecture that outperforms character-level models while being twice faster in training and 20% faster in inference. The proposed pipeline yields better performance than character-level conversational models according to human assessor testing.

Keywords: morphological agreement, conversational models, morphology, natural language processing

1. Introduction

Conversational models appear in a wide range of applications, from simple rule-based chatbots to complex personal assistants. Over recent years, neural conversational models (Vinyals and Le, 2015) have almost entirely replaced classical approaches based on information retrieval (Jafarpour and Burges, 2010). Neural models usually operate on the word level and require plenty of data and computational resources for training and inference. For fusional languages, the vocabulary size becomes a bottleneck for both memory and computation: we have to learn and store an embedding vector for each word and all its morphological forms. Several vocabulary reduction techniques, such as n-gram level models or byte pair encodings (Sennrich et al., 2015), have been proposed to mitigate this problem and reduce the vocabulary. Nevertheless, subword-level vocabularies for neural conversational models usually lead to inferior performance in practice, in part due to inconsistencies in the generated text: the model not only has to interpret the input question but also to formulate and spell the generated answer correctly. In practice, character-level models seem to produce only frequent words, ignoring rare ones due to the risk of misspelling them.

© 2018 D. Polykovskiy, D. Soloviev & S. Nikolenko.

Another technique to reduce the vocabulary size is to lemmatize all words prior to building the vocabulary. In this case, all word forms of a given word are merged into a single token. For languages with rich morphology, lemmatization can greatly decrease the vocabulary size and make the model computationally feasible. While word-level models on normalized vocabularies can produce diverse, detailed answers, the resulting texts are not yet ready to go to the end user since they are not grammatically correct. In this work, we propose to train an auxiliary model to perform *morphological agreement*, i.e., map a sentence composed of lemmatized words back to their proper morphological forms. A conversational model on a lemmatized vocabulary with subsequent morphological agreement provide a new pipeline for conversational systems with reduced vocabulary size. Thus, our contribution is twofold: we propose a neural network architecture to perform morphological agreement in languages with rich morphology and introduce a new approach to building conversational models based on generating normalized text and then performing morphological agreement with the proposed model.

The paper is organized as follows. In Section 2, we present neural approaches to the morphological agreement problem: a sequence-to sequence character-based model (Section 2.1), a neural network architecture for morphological agreement (2.2), and the full conversational pipeline (Section 2.3). Section 3 presents experimental evaluation with experiments in morphologically rich languages, Section 4 presents related work, and Section 5 concludes the paper.

2. Neural Morphological Agreement

We begin by defining the morphological agreement problem. Consider a grammatically correct sentence S, represented as a sequence of words $w_1, w_2, \ldots, w_K$. Let $\mathcal{L}(w)$ be a function that maps a word to its standard form, e.g., $\mathcal{L}(\text{“went”}) = \text{“go”}$. The morphological agreement problem is to learn an inverse of $\mathcal{L}$ applied to complete sentences—a mapping that restores $w_1, w_2, \ldots, w_K$ from $\mathcal{L}(w_1), \mathcal{L}(w_2), \ldots, \mathcal{L}(w_K)$. Lemmatization itself is a well-known problem (Chrupala, 2006); it can be solved reasonably well even with a simple dictionary lookup, and modern context-sensitive approaches to lemmatization based on conditional random fields (CRF) or similar models achieve excellent results (Müller et al., 2015). The inverse task of learning $\mathcal{L}^{-1}$, however, is much harder and much more context-dependent.

2.1. Sequence-to-sequence character-based model

A straightforward approach to the morphological agreement problem is to train a character-based sequence-to-sequence model (Sutskever et al., 2014b) to map a normalized sentence

$$S^N = \mathcal{L}(w_1), \mathcal{L}(w_2), \ldots, \mathcal{L}(w_K) \tag{1}$$

represented as a single string to the original sentence $S = w_1, w_2, \ldots, w_K$. A commonly used sequence-to-sequence architecture contains two recurrent networks: encoder E and decoder D. The encoder generates an embedding vector $E(S^N)$ from the string S^N, character by character. The decoder then uses the generated embedding to maximize the probability of the correct sentence S:

$$\max_{E,D} \mathbb{E}_{S \sim p_{\text{data}}} \log P_D\left(S | E(S^N)\right). \tag{2}$$

While this model does yield a relevant baseline, this approach has two drawbacks. An encoder has to compress all relevant information from S^N into a single embedding vector of fixed size,

leading to poor performance on long sentences (see Section 3.3). Also, the generative (decoding) procedure does not even guarantee that the produced sentence will have the same number of words as the input sentence. While this constraint seems rather mild, it turns out that incorporating the knowledge about the correct number of words can improve the overall quality of the model.

Attention mechanisms (Luong et al., 2015; Yao et al., 2015) are often used to reduce the payload of the embedding vector. For recurrent encoder and decoder architectures, attention operates by computing a weighted sum of intermediate encoder outputs for each input token. The resulting vector serves as an additional feature for the decoder at each time stamp. Weights in the sum are computed dynamically by the decoder and can depend on the input. As a result, a neural architecture with attention partially avoids the embedding bottleneck, but it still has to pack a lot of information in this vector and still performs poorly on longer sequences.

2.2. Concorde

To solve both issues with sequence-to-sequence models described above, we propose a neural network model specifically designed for the morphological agreement task. The main idea of the proposed model is to decompose morphological agreement for the entire sentence into several tasks of restoring individual words w_i from their normalized forms $\mathcal{L}(w_i)$. To do so, we assume that for each word in a sentence we can recover its *morphological context* Z_i, a vector containing all relevant information about morphology, including tense, case, and plurality. Given this vector and a normalized sentence, all target words w_i become conditionally independent:

$$P(S|Z, S^N) = \prod_{i=1}^{|S|} P\left(w_i|Z_i, \mathcal{L}(w_i)\right). \tag{3}$$

Specifically, we obtain vectors Z_i by the following procedure. First, we compute embeddings of normalized words using a character-based word encoder E^w. We then pass these embeddings (one per word) through a bidirectional LSTM (Graves et al., 2013), producing vectors Z_i; note that by using bidirectional LSTMs, we allow Z_i to capture both left and right context of a word, drawing upon information about the whole sentence. Finally, probabilities $P(w_i|Z_i, \mathcal{L}(w_i))$ are modeled with a recurrent character-based decoder D^w using a context Z_i as an initial hidden and cell state along with attention weights over the corresponding normalized word characters $\mathcal{L}(w_i)$.

Formally, the model and the corresponding optimization problem can be written as follows:

$$\max_{Z, E^w, D^w} \mathbb{E}_{S\sim p_{\text{data}}} \sum_{i=1}^{|S|} \log P(w_i|\mathcal{L}(w_i), Z_i) \tag{4}$$

We call the proposed model *Concorde* and illustrate it in Figure 1. The Concorde model solves both issues with naive sequence-to-sequence models discussed above. The information that was previously contained in a single embedding vector can now be distributed among multiple embeddings, one for each word in the sentence. Furthermore, this embedding only has to contain information about characters of a single word, which is generally a short sequence, significantly improving the performance on long sentences, as we will see in Section 3.3. Unlike sequence-to-sequence models, Concorde is constrained to generate the correct number of words, as the decoder is evaluated separately for each input word. The model is also parallelizable: while sequence-to-sequence models have to process each token sequentially, Concorde can perform encoding and

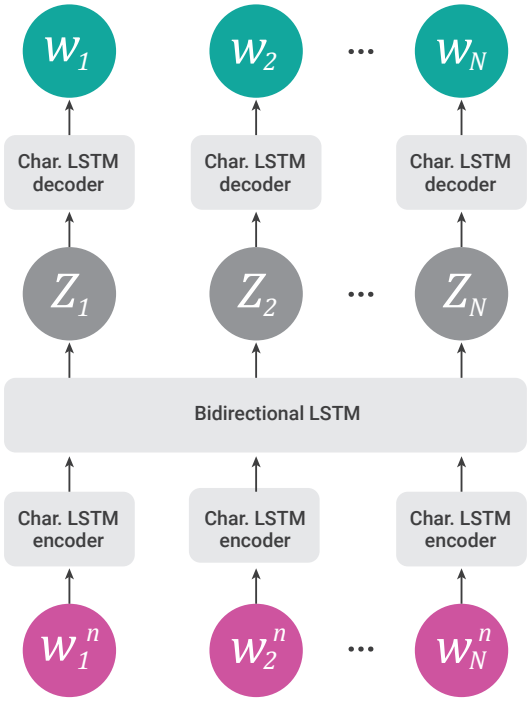

Figure 1: Concorde model.

decoding for each word in parallel. The only sequential part is the bidirectional LSTM that processes only $|S|$ elements (words) for each sentence.

2.3. Scaling up to neural conversation: Q-Concorde

The proposed Concorde model is a general framework for solving the morphological agreement task. One possible application of Concorde is to build conversational models. Many languages have rich morphology, with numerous word forms for many words. Storing all word forms in the dictionary significantly increases vocabulary size, which makes training conversational models computationally expensive and requires the training set to contain multiple instances of each word form. We propose to reduce vocabulary size by lemmatizing all words. Such a model can still produce meaningful answers, but the generated texts require post-processing to turn them into grammatically correct forms, and we propose to use the Concorde model for this post-processing, i.e., for morphological agreement. The model pipeline operates as follows: a user asks a question, words of this question are normalized and used as input for a conversational model, the conversational model produces a normalized answer, and finally this answer is transformed into a grammatically correct sentence using the Concorde model.

This pipeline does not use information about the question's morphology, leading to various mistakes. For example, lemmatization loses information about gender, and a morphological agreement cannot infer the correct gender. To solve this issue, we modify the Concorde model to by adding morphological features from the original, grammatically correct question. We extract relevant

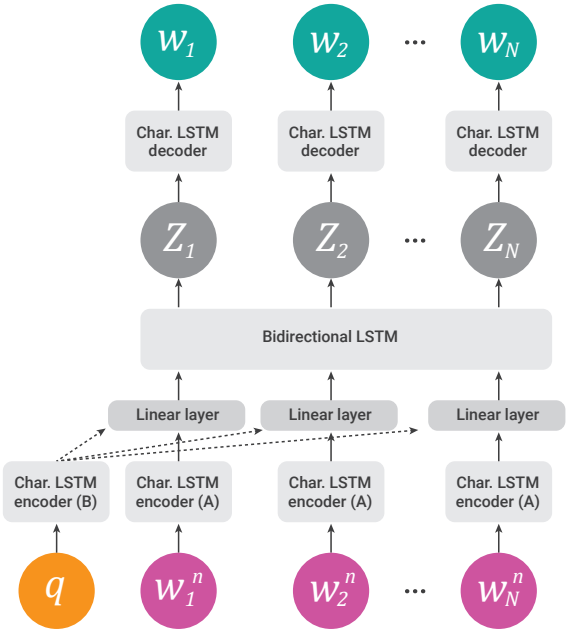

Figure 2: Q-Concorde model.

information from the question through a character-based encoder Q. It produces an embedding vector which we concatenate with embeddings of all words of the normalized answer. We then map obtained embeddings through a linear layer to the original embedding dimension (see Figure 2). We call this model *Q-Concorde*.

3. Evaluation

In this section, we present the our experimental results. We evaluate Concorde and Q-Concorde in two steps. First we compare the performance of Concorde and previously developed character-level models on the morphological agreement task in three languages: French, Polish, and Russian. We then evaluate the Q-Concorde model on a question answering task based on natural language dialogues.

3.1. Experimental setup

We constructed training sets by applying lemmatization to texts in three natural languages. For French and Polish, we used large lemmatization vocabularies (Měchura, 2018), and for Russian we used the `pymorphy2` library (Korobov, 2015). Both Concorde and Q-Concorde models were designed as a two-layer LSTM encoder and decoder with a hidden size of 512.

We compared our model to two baselines: unigram CharRNN and a hierarchical model. The unigram model is a standard sequence-to-sequence model with attention (Luong et al., 2015) that operates with characters or pairs of characters as tokens. In the CharRNN models, we used a 2-layer LSTM as the encoder, and the decoder consisted of a 2-layer LSTM followed by an attention layer

Table 1: Comparison of four morphological agreement models for three different languages: word accuracy (**WA**) and sentence accuracy (**SA**).

	French		**Russian**		**Polish**	
	WA	**SA**	**WA**	**SA**	**WA**	**SA**
CharRNN	83.10	50.50	75.88	44.97	70.41	38.02
Hierarchical	83.03	50.52	77.73	46.26	70.71	38.31
Concorde	**87.90**	**56.86**	**85.18**	**53.97**	**79.76**	**44.93**

and another recurrent layer. The third baseline was a hierarchical model motivated by Johansen et al. (2016): we first embed each word using a recurrent encoder and then compute the sentence embedding by running a word-level encoder on these embeddings. For baselines, the hidden size was 768, resulting in a comparable number of parameters for all models.

We trained our models with the Adam optimizer (Kingma and Ba, 2014) with batch size 16 and initial learning rate 0.0002, halving it after each $50{,}000$ updates. We terminated training after $300{,}000$ updates, which was sufficient for convergence in our experiments.

3.2. Word and sentence accuracy

We used corpora from the OpenSubtitles database Lison and Tiedemann (2016) for French, Russian, and Polish languages. We performed morphological agreement for each subtitle independently. The vocabulary contained words that appeared more than 10 times among the first 10 million tokens, and we removed sentences longer than 20 words.

We evaluated our model with respect to two metrics: *word accuracy* and *sentence accuracy*. Word accuracy is the fraction of words that were correctly generated by the model. Sentence accuracy is the fraction of sentences that were reconstructed with no mistakes. Table 1 summarizes our results on the OpenSubtitles dataset: out of four models, Concorde consistently yields the best performance across all languages, while the hierarchical model places second.

We also manually inspected our model to compare its performance in different cases. Table 2 shows some examples where Concorde was able to infer the plural form and gender for unseen words. For the Russian language, we have found that the model was able to learn some rather nonstandard rules such as changing the letter “я” to “й” in the middle of the stem in the plural form for some words: “один заяц” (one hare) becomes “два зайца” (two hares) in plural form rather than “два заяца” as might be naively suggested from the majority of other examples, and the model picks up on this morphological exception without ever seeing any hares in the training set.

Results shown in Table 3 indicate that Concorde is also able to infer gender from words. To show that we chose feminine, masculine, and neuter words and used the model to predict agreement with the word “one”, as in “one plate” or “one pie” because “one” changes by gender in all three languages. As we see in Table 3, the model can indeed solve this task and assign the correct gender to the numeral.

Table 4 shows results on full sentences. The Russian example, translated literally as “The girl Alice lives in the adjacent entrance” (“The girl Alice lives next door”) is quite complex for agreement:

Table 2: Singular and plural forms.

	Singular		Plural	
	Normalized	**Output**	**Normalized**	**Output**
Fr	un château	Un château	deux château	Deux châteaux
Ru	один заяц	Один заяц	два заяц	Два зайца
Pl	jeden ołówek	Jeden ołówek	dwa ołówek	Dwa ołówki

Table 3: Gender agreement: **f.** — feminine, **m.** — masculine, **n.** — neuter.

		Normalized	**Output**
French	**f.**	un personne	Une personne
	m.	un témoin	Un témoin
Russian	**f.**	один тарелка	Одна тарелка
	m.	один пирог	Один пирог
	n.	один растение	Одно растение
Polish	**f.**	jeden kawa	Jedną kawa
	m.	jeden człowiek	Jeden człowiek
	n.	jeden mleko	Jedno mleko

e.g., to choose the correct form for the word "соседнем" ("adjacent"), the network had to use multiple markers from different parts of the sentence: gender from "подъезд" ("entrance") and case from "в" ("in"). Results of our manual inspection also indicate that the model works well even on hard examples that contain words not from the dataset.

3.3. Performance on long sentences

As a motivation for our model, we argued that making shorter input-output paths may reduce the information load on the embedding vector and thus improve long connections. To test this hypothesis, we computed average sentence accuracy for different input lengths; Fig. 3 shows sample results for the French language. The plots clearly show that the performance of all baseline models significantly deteriorates as the input length increases. This is, however, not the case for Concorde: while charRNN models drop to virtually 0% accuracy when the input is 100 characters long, Concorde still yields about the same performance as for short sentences. This result can be explained by the way in which models use embedding vectors. Baseline models have to share the embedding's capacity among all words in a sentence. Concorde, on the other hand, has a separate embedding for each word and does not need to squeeze the whole sentence into a single vector.

Character-level models perform better for short sentences (about 33% of the test set). This is probably because the embedding vector has sufficient capacity in these cases. Despite worse performance for short inputs, Concorde still handles many important cases very well, including those discussed in Section 3.2. Note that this means that one can easily improve upon the overall accuracy by using CharRNN for short sentences and Concorde for longer ones: in our experiments, this has increased sentence accuracy from **56.86%** (for Concorde; see Table 1) to **62.95%** for French, from **44.93%** to **52.61%** for Polish, and from **53.97%** to **61.67%** for Russian.

Table 4: Characteristic examples: **N** — normalized, **O** — model output.

Fr	**N**	il vouloir d l amour d le joie de le bon humeur
	O	Je veux d l amour d la joie de la bonne humeur
Ru	**N**	девочка элиса жить в соседний подъезд
	O	Девочка Элис живет в соседнем подъезде
Pl	**N**	on dostać dużo oda nikt inny
	O	Nie dostaniesz więcej od nikogo innego

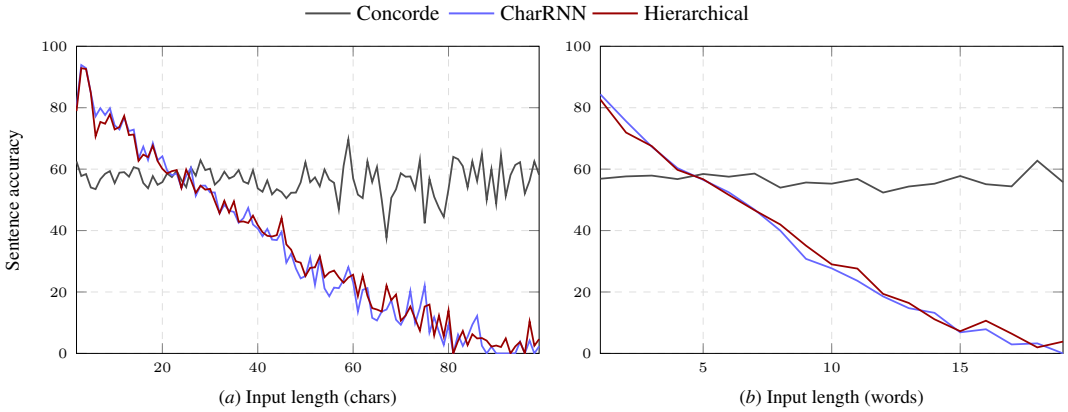

Figure 3: Sentence accuracy as a function of sentence length (French).

3.4. Dialogue modeling with Q-Concorde

We also evaluated the proposed two-step conversational model—generate the normalized answer and then apply Concorde to produce a grammatically correct text—on a corpus of question-answer pairs that we obtained from `otvet.mail.ru` web site, a Russian-language service for questions and answers similar to Quora. Unlike other available datasets, our corpus contains general knowledge questions where the trained model has to answer questions about movies, capitals, and other common trivia. This requires the vocabulary to contain many words related to rare entities, making the vocabulary without normalization extremely large.

We compared the Q-Concorde and Concorde models to show that Q-Concorde can indeed grasp important morphological features from the context. We also trained baseline models with context concatenated to the input sentence (with a special delimiter in between). Table 5 shows word and sentence accuracies: Concorde outperforms baselines even though it does not have access to the context, and Q-Concorde further improves upon Concorde.

We inspected some cases where Q-Concorde shows better performance than Concorde (Table 6). In the first example, the question deals with a single object, and Q-Concorde correctly used a singular form while Concorde used plural. Q-Concorde also successfully carries the correct case (example 2) and time (example 3) from the question. On the other hand, some mistakes made by Q-Concorde are shown in Table 7: in the first example, Q-Concorde was not able to decide whether to use the polite form (plural “you” or singular “thou”) and used a singular verb with plural possessive pronoun, breaking agreement.

Table 5: Performance of Q-Concorde and baselines on otvet.mail.ru dataset.

Model	Word Accuracy, %	Sentence Accuracy, %
CharRNN	72.83	38.88
Hierarchical	74.99	35.2
Concorde	81.51	43.75
Q-Concorde	**83.13**	**48.10**

Table 6: Cases where Q-Concorde is **better** than Concorde.

1	**Question**	какой ты человек сложный или с тобой все просто
	Conc.	простые и добрые
	Q-Conc.	простой и добрый
2	**Question**	что прячут под вуалью
	Conc.	тайна
	Q-Conc.	тайну
3	**Question**	вас матери сколько носили месяцев
	Conc.	по стандарту выносят
	Q-Conc.	по стандарту выносили

Q-Concorde can generate different replies depending on the lexical features of the question. For example, Table 8 shows grammatical changes in the question and the corresponding answers: we changed the tense from present to past ("what are you doing?" vs. "what did you do?") and gender from masculine to feminine, and the answer used the correct morphological forms in each case (although in the third example the grammar turned out to be slightly off precisely due to these changes).

Finally, we applied Q-Concorde in the proposed pipeline for training conversational models. We compared our model with a 3-layer character-level sequence-to-sequence model trained on grammatically correct sentences. To generate diverse answers, we trained two models: one to predict the answer given a question and another to predict the question given its answer, as suggested in (Li et al., 2015). This allows us to discard answers that are too general.

To compare the two models, we set up an experimental environment where assessors were asked to choose one of two possible answers to a given question: one generated by the character-based model and the other generated by our pipeline, shown in a random order. In **62.1%** of the cases the assessors selected the proposed model, preferring the answers from charRNN in the other **37.9%**.

3.5. Computational costs

The time needed to process a batch of sentences is much higher for character-level models since they need to process longer sequences sequentially. Table 9 reports the running time for forward and backward passes of one batch (16 objects) and other important computational characteristics, measured on a GeForce GTX TITAN X GPU. It turns out that Concorde and Q-Concorde have comparable inference time, but train faster than unigram and hierarchical models.

Table 7: Cases where Q-Concorde is **worse** than Concorde.

1	**Question**	какой тарифный план выбрать
	Concorde	позвоните вашему оператору
	Q-Concorde	позвони вашему оператору
2	**Question**	что у вас на завтрак
	Concorde	только поцелуи
	Q-Concorde	только поцелуй

Table 8: Examples of Q-Concorde in action.

Question	**Norm. answer**	**Answer**
чего ты делаешь		я хожу гулять
чего ты делала	я ходить гулять	я ходила гулять
чего ты делал		я ходил гулял

4. Related work

Dialogue and conversation modeling are a characteristic example of *sequence-to-sequence* problems: given a sequence of words and/or symbols, the model has to produce a reasonable reply, i.e., another sequence of words/symbols. The neural conversational model introduced by Vinyals and Le (2015) uses the *seq2seq* framework from (Sutskever et al., 2014a); see Figure 4 for an illustration. This direct *seq2seq* approach can be easily extended to many applications, including machine translation and question answering, but unlike machine translation in this case it cannot really be expected to model the dialogue since human dialogue usually carries over the context for a very long time, pursuing long-term goals that probably cannot be modeled within *seq2seq*. Still, experiments in (Vinyals and Le, 2015) show very reasonable dialogues both in the IT helpdesk context and in the general context of movie subtitles (Tiedemann, 2009).

Serban et al. (2016) extend the hierarchical recurrent encoder decoder architecture (HRED) proposed by Sordoni et al. (2015), who used it for context-aware query suggestion for information retrieval. The basic idea of (Serban et al., 2016) is to view dialogue as a two-level system: a sequence of utterances, each of which is in turn a sequence of words. To model this two-level system, HRED trains (1) an *encoder* RNN that maps each utterance in a dialogue into a single utterance vector; (2) a *context* RNN that processes all previous utterance vectors and combines them into the current context vector; (3) a *decoder* RNN that predicts the tokens in the next utterance, one at a time, conditional on the context RNN (see Figure 5). Serban et al. (2016) use bidirectional RNNs, initialize the weights with *word2vec* representations trained on a large dataset (Google News), bootstrap from a question-answer subtitle corpus with short questions and answers, and perform the main training with the MovieTriples dataset based on the Movie-DiC dataset (Banchs, 2012). Continuing this, Serban et al. (2017a,b) developed a variational lower bound for the hierarchical model and optimizes it; the resulting *Variational Hierarchical Recurrent Encoder-Decoder* (VHRED) model estimates latent variables in the dialogue that model the complex dependencies between individual utterances, and extended the model to piecewise constant priors, which leads to multimodal document modeling, generating responses to the time and events in the original query.

Table 9: Computational comparison: time per batch and GPU memory for training and inference.

	Uni	**Hier**	**Concorde**	**Q-Conc.**
# params	26M	36M	32M	35M
Training	410 ms	379 ms	190 ms	285 ms
	2316M	1970M	4290M	4512M
Inference	153 ms	146 ms	128 ms	147 ms
	1417M	2025M	3419M	4445M

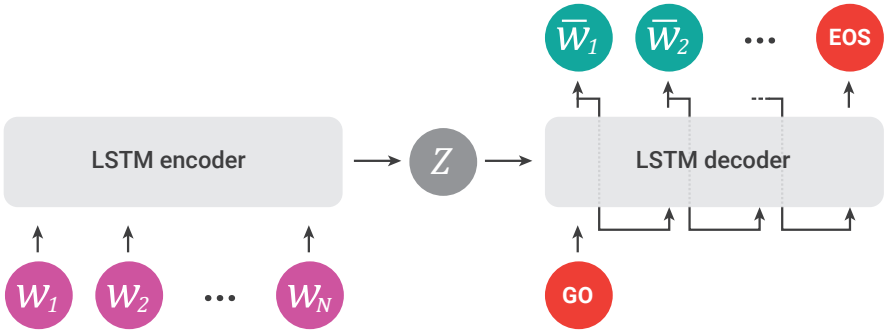

Figure 4: The *seq2seq* conversational architecture (Vinyals and Le, 2015).

Other recent developments apply reinforcement learning to improve conversational models, both directly for response generation (Li et al., 2016b) and with online active reward learning (Su et al., 2016), model intentionality in dialogue (Yao et al., 2015) or directly model the events of relaying relevant information (Wen et al., 2016), add extra latent variables to the models to model *personas*, so that the dialogue can be more consistent (Li et al., 2016a), improve diversity of responses (Cao and Clark, 2017), and so on. While it is still a long way to go before actual general-purpose dialogue, conversational models are a rapidly developing field.

We know of no direct attempts to add morphology to conversational models in a way similar to Concorde; most works concentrate on English or Chinese languages. Special provisions for morphology-rich languages have been made, however, in other fields of natural language processing, usually with character-based models. For example, Ballesteros et al. (2015) improve upon a parser based on stack LSTMs (Dyer et al., 2015) with bidirectional LSTMs producing character-based word representations; Chung et al. (2016) construct a machine translation model that augments word embeddings with a character-level model, and Google's Neural Machine Translation system breaks words up into pieces (Wu et al., 2016).

Word inflection is a similar task, but word inflection models generate specified word forms, while in Concorde the model has to automatically choose the correct form. Durrett and DeNero (2013) propose a supervised approach to predicting the set of all word forms by generating transformation rules from known inflection tables, using conditional random fields (CRF) for unseen base forms. Aharoni et al. (2016) and Faruqui et al. (2015) use bidirectional LSTMs to encode the word; Faruqui et al. (2015) add different decoders for different word forms, while Aharoni et al. (2016) suggest to have a single decoder and attach morphological features to its input. Apart from RNNs, convolutional networks (CNNs) have been applied in (Ostling, 2016; Faruqui et al., 2015), where raw text data

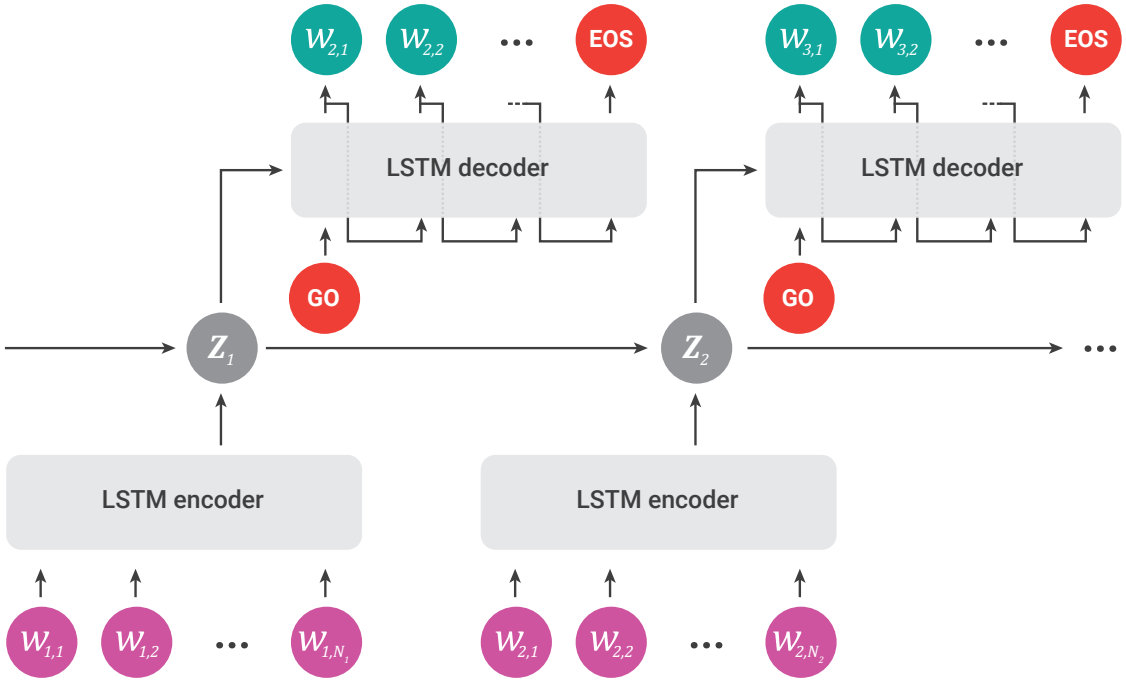

Figure 5: The HRED architecture for a conversational model (Serban et al., 2016).

first passes through convolutional layers and then goes to a recurrent encoder. Kim et al. (2016) use character-level encoder and word-level decoder.

In a way, Concorde continues and extends this line of work to conversational models, but we attack it from a different angle, explicitly separating the problems of producing a good response as a sequence of words and achieving morphological agreement with this sequence. The morphological agreement problem has not been well studied in literature. We note the works (Linzen et al., 2016; Bernardy and Lappin, 2017) that use RNNs to learn syntactic agreement; however, they solve a more specific problem, e.g., predict the number or tense of a specific verb in a sentence where all other words are already in their correct forms, while Concorde attempts to reconstruct all correct forms at once. Thus, to the best of our knowledge Concorde is a novel approach to conversational models and, moreover, it introduces a novel useful task of full-scale syntactic agreement.

5. Conclusion

In this work, we have considered the conversational modeling problem for fusional languages, where vocabularies become infeasible due to different morphological forms. We have proposed a novel pipeline for conversational models where the model first generates normalized replies and then reconstructs proper agreement, saving memory by reducing vocabulary size with lemmatization. We have introduced a neural network model, called Concorde, for the morphological agreement problem that outperforms conventional sequence-to-sequence models on this task, showing that Concorde significantly outperforms character-based models for morphological agreement. We have also shown that the proposed conversational pipeline outperforms other conversational models while being faster in both training and inference.

Acknowledgments

The work was performed by members of Neurolab at Mail.ru group. The modification of Concorde — Q-Concorde was also supported by the Russian Science Foundation Grant 17-71-20072.

References

Roee Aharoni, Yoav Goldberg, and Yonatan Belinkov. Improving sequence to sequence learning for morphological inflection generation: The biu-mit systems for the sigmorphon 2016 shared task for morphological reinflection. *ACL 2016*, page 41, 2016.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proc. EMNLP 2015*, pages 349–359, Lisbon, Portugal, 2015. ACL.

Rafael E. Banchs. Movie-dic: A movie dialogue corpus for research and development. In *Proc. 50th ACL: Short Papers - Volume 2*, ACL '12, pages 203–207, Stroudsburg, PA, USA, 2012. ACL.

Jean-Philippe Bernardy and Shalom Lappin. Using deep neural networks to learn syntactic agreement. *Linguistic Issues in Language Technology*, 15:1–15, 2017. ISSN 1945-3590.

Kris Cao and Stephen Clark. Latent variable dialogue models and their diversity. *EACL*, 2017.

Grzegorz Chrupala. Simple data-driven context-sensitive lemmatization. *Procesamiento del Lenguaje Natural*, 37, 2006.

J. Chung, K. Cho, and Y. Bengio. A character-level decoder without explicit segmentation for neural machine translation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1693—-1703, 2016.

Greg Durrett and John DeNero. Supervised learning of complete morphological paradigms. In *HLT-NAACL*, pages 1185–1195, 2013.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. Transition-based dependency parsing with stack long short-term memory. In *Proc. 53rd ACL and the 7th IJCNLP, Vol. 1: Long Papers*, pages 334–343, Beijing, China, 2015. ACL.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. Morphological inflection generation using character sequence to sequence learning. *arXiv preprint arXiv:1512.06110*, 2015.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

Sina Jafarpour and Chris J.C. Burges. Filter, rank, and transfer the knowledge: Learning to chat. *Microsoft Technical Report*, July 2010.

Alexander Rosenberg Johansen, Jonas Meinertz Hansen, Elias Khazen Obeid, Casper Kaae Sønderby, and Ole Winther. Neural machine translation with characters and hierarchical encoding. *arXiv preprint arXiv:1610.06550*, 2016.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *AAAI*, pages 2741–2749, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, 2014.

Mikhail Korobov. Morphological analyzer and generator for russian and ukrainian languages. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 320–332. Springer, 2015.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *In. Proc of NAACL*, 2015.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003. Association for Computational Linguistics, 2016a. doi: 10.18653/v1/P16-1094.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In *Proc. 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1192–1202, 2016b.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association of Computational Linguistics*, 4:521–535, 2016.

Pierre Lison and Jörg Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. *European Language Resources Association*, 2016. URL http://opus.lingfil.uu.se/OpenSubtitles2016.php.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *EMNLP*, 2015.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, 2015.

Michal Měchura. Machine-readable lists of lemma-token pairs in 23 languages. *GitHub*, 2018. URL https://github.com/michmech/lemmatization-lists/.

Robert Ostling. Morphological reinflection with convolutional neural networks. *ACL 2016*, page 23, 2016.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3776–3783. AAAI Press, 2016.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proc. 31st AAAI*, pages 3295–3301, 2017a.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301, 2017b.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proc. 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 553–562, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806493.

Pei-Hao Su, Milica Gasic, Nikola Mrkši, Lina M. Rojas Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proc. 54th ACL, Vol. 1: Long Papers*, pages 2431–2441, Berlin, Germany, August 2016. ACL.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014a.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014b.

Jorg Tiedemann. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing (vol V)*, pages 237–248. John Benjamins, Amsterdam/Philadelphia, 2009.

Oriol Vinyals and Quoc Le. A neural conversational model. *ICML Deep Learning Workshop*, 2015.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve J. Young. Conditional generation and snapshot learning in neural dialogue systems. In *Proc. 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2153–2162, 2016.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*, 2016.

Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. Attention with intention for a neural network conversation model. *NIPS Workshop*, 2015.