Improving Named Entity Recognition in Spoken Dialog Systems
by Context and Speech Pattern Modeling

By

BINH MINH NGUYEN
THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Zhou Yu, Chair

Premkumar Devanbu

Hao-Chuan Wang

Committee in Charge

2021

Abstract

While named entity recognition (NER) from speech has been around as long as NER from written text has, the accuracy of NER from speech has generally been much lower than that of NER from text. The rise in popularity of spoken dialog systems such as Siri or Alexa highlights the need for more accurate NER from speech because NER is a core component for understanding what users said in dialogs. Deployed spoken dialog systems receive user input in the form of automatic speech recognition (ASR) transcripts, and simply applying NER model trained on written text to ASR transcripts often leads to low accuracy because compared to written text, ASR transcripts lack important cues such as punctuation and capitalization. Besides, errors in ASR transcripts also make NER from speech challenging. We propose two models that exploit dialog context and speech pattern clues to extract named entities more accurately from open-domain dialogs in spoken dialog systems. Our results show the benefit of modeling dialog context and speech patterns in two settings: a standard setting with random partition of data and a more realistic but also more difficult setting where many named entities encountered during deployment are unseen during training.

Contents

1 Introduction

1.1 Background

Named entity recognition (NER) is the task of extracting proper names of people, locations, and so on from text or speech (Grishman and Sundheim, 1996). There has been a lot of work on NER from written text with many systems achieving impressive results (Devlin et al., 2019; Akbik, Bergmann, and Vollgraf, 2019). Although, NER from speech has been around for the same time as NER from text (starting with work by Kubala et al. (1998)), accuracy of NER from speech still lags behind the accuracy of NER from text. The rise in popularity of spoken dialog systems such as Siri or Alexa highlights the need for more accurate NER from speech because NER is a core component for understanding what users said in dialogs. In spoken dialog systems, humans interact with the systems using natural speech to accomplish certain tasks (task-oriented dialog) or just to be entertained (chit-chat or open-domain dialog) (Jurafsky and Martin, 2009). These systems require speech transcripts as input in real-time and the transcripts are obtained using automatic speech recognition (ASR) components (Turmo et al., 2009).

Much previous work on NER from speech data, such as broadcast news, applied text-based NER systems to the output of an ASR system (Palmer and Ostendorf, 2001). However, NER performance degraded significantly (20 points drop in F1 score) when applying a NER trained on written data to transcribed speech (Kubala et al., 1998). This could be because applying text-based NER system to ASR output ignores the differences in styles and conventions in written and spoken language (Palmer and Ostendorf, 2001). For example, spoken utterances in spontaneous speech are usually much shorter than written prose so the utterances could be ambiguous when taken out of

context. In addition, speech also contains disfluencies, repetitions, restarts and corrections (Turmo et al., 2009). Besides, text-based NER system may depend on cues such as sentence punctuation and capitalization which are not present in ASR transcripts (Shriberg et al., 2000). Furthermore, ASR is not error-free and errors in ASR transcripts lead to cascading errors in NER (Turmo et al., 2009). Due to factors such as greater variation in speakers, greater variation in content because of the open-ended nature of open-domain dialogs, and less professional recording environment, ASR transcripts from spoken dialog systems often contain more errors than that from broadcast news, making NER in dialogs a much more challenging task.

1.2 Related Work

Recent NER models perform well on clean text datasets such as CoNLL (Tjong Kim Sang and De Meulder, 2003) and OntoNotes (Hovy et al., 2006), but less well on noisy data (Mayhew, Gupta, and Roth, 2020) such as the WNUT dataset (Derczynski et al., 2017). In term of F1 score, the current state-of-the-art model (Akbik, Bergmann, and Vollgraf, 2019) achieves 93% on the CoNLL dataset but only 49% on the WNUT dataset. The overreliance of NER models on the convention of capitalizing named entities (Derczynski et al., 2017) partly explains why they perform poorly on text where capitalization is absent or noisy. In spoken dialog systems, inputs to NER models are ASR transcripts which not only lack capitalization and punctuation but also contain transcription errors (Sundheim, 1995; Lenzi, Speranza, and Sprugnoli, 2012). Although, joint decoding of ASR transcript and NER output (Caubrière et al., 2020) partly lessens the impact of ASR errors on NER, detecting named entities in ASR transcripts remains a challenging problem (Galibert et al., 2014).

Prior work on NER from ASR transcripts focus on reducing ASR errors (Palmer and Ostendorf, 2001), exploiting multiple ASR hypotheses (Horlock and King, 2003; Béchet et al., 2004), or exploiting additional information such as speech pattern features (Katerenchuk and Rosenberg, 2014). Examples of speech pattern features are ASR confidence (Sudoh, Tsukada, and Isozaki, 2006), pauses, and word durations (Hakkani-Tür et al., 1999). Recently, Cervantes and Ward (2020)

used solely prosidic speech features to spot location mentions. Our work is similar to Katerenchuk and Rosenberg (2014) in that we also utilize speech pattern features. However, while Katerenchuk and Rosenberg (2014) focused on broadcast news speech, our work focuses on spoken dialogs. Thus, besides speech pattern features, our models also exploit dialog context for more accurate NER. In addition, Katerenchuk and Rosenberg (2014) used a separate classifier trained on data from a small set of speakers to derive speech pattern features, so the predicted features may not generalize to more diverse populations. In contrast, our approach is more integrated since the speech pattern features encoder is part of the proposed models thereby encouraging the models to learn features that are more generalizable.

1.3 Contribution

We propose two models that exploit dialog context and speech patterns which are available in open-domain dialogs from spoken dialog systems to achieve more accurate NER. Our results show the benefit of modeling dialog context and speech patterns in two settings: a standard setting with random partition of data and a more realistic but also more difficult setting where there is little overlap between named entities during training and testing.

2 Methods

2.1 Motivation

Dialog utterances are usually short and ambiguous when taken out of context, therefore identifying named entities in dialog utterances can be challenging. Figure 2.1 shows two challenging cases where dialog context and speech patterns can aid NER. Although users' utterances are simi-

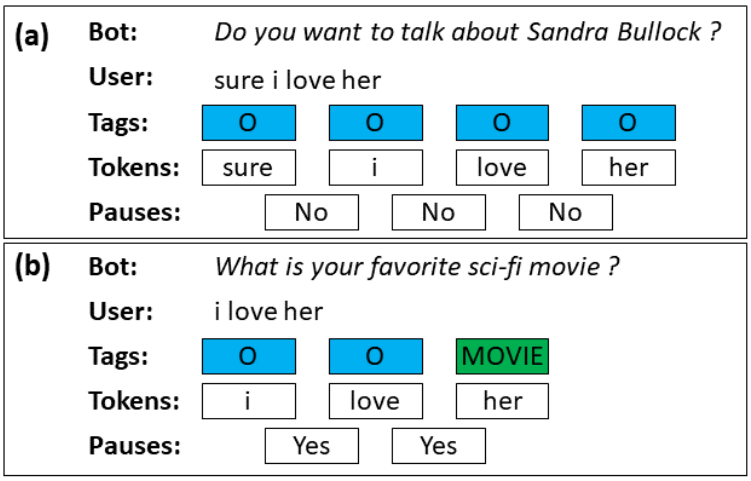

FIGURE 2.1: Dialog context and speech patterns help distinguishing "*her*" in (a) is a mentioned pronoun and "*her*" in (b) is a named entity (the 2013 sci-fi movie Her). Examples are not actual interaction data.

lar, the phrase "*her*" is a named entity in the second case but not in the first case. Without knowing what the bot said (i.e. dialog context), the best guess is that "*her*" refers to a person and therefore not a named entity. However, when "*i like her*" is a response to the question "*What is your favorite sci-fi movie?*", "*her*" is a named entity (the 2013 sci-fi movie Her). Although users usually mention their favorite movies when asked, they can also change topic, making contextual NER non-trivial.

Thus, exploiting dialog context could help resolving named entities in users' utterances in more difficult cases.

Besides context, speech pattern features, which include prosodic and non-prosodic features (Shriberg et al., 2000), might also help identifying named entities. In particular, pauses' duration, words' duration, and tokens' ASR confidence are some readily available features that may be useful for NER. Pauses might occur when speakers were choosing their words (Goldman-Eisler, 1958), so pauses might indicate subsequent named entities in utterances. Figure 2.1b shows the user pausing prior to uttering the named entity "*her*" as the user might have been considering different named entities. In contrast, in Figure 2.1a, there was no pause probably because the user was saying a set phrase so there was no difficult choice involved. Furthermore, pauses could signal boundaries (punctuation) between grammatical structures within utterances (Reich, 1980; Chen, 1999). Since punctuation is an important feature in NER (Nadeau and Sekine, 2007) and punctuation is missing in ASR transcript, pauses could potentially replace the missing punctuation. Exaggerated variation in word durations and pauses could be present when pronouncing non-native names (Fitt, 1995; Rangarajan and Narayanan, 2006). Tokens' confidence might also predict the presence of named entities since named entities appear less often than other words in ASR training data. Tokens' confidence have been used previously in NER task (Palmer and Ostendorf, 2001; Sudoh, Tsukada, and Isozaki, 2006).

2.2 Model

We propose two NER models for dialog which take a dialog exchange as input. A dialog exchange consists of a bot's utterance followed by an user's utterance, and the models must label named entities in the user's utterance, taking into account the context (the bot's utterance). The user's utterance includes lexical features (i.e. word tokens or word pieces) and speech pattern features which are pauses' duration, words' duration, and tokens' ASR confidence. Both models have three components: (1) a context encoder, (2) a speech pattern encoder, and (3) a sequence tagger. The

context encoder and speech pattern encoder are the same in both models and the encoders provide additional clues for the sequence tagger to accurately label named entities. The first model's sequence tagger is a widely used model for NER from written text based on BiLSTM-CRF (Ma and Hovy, 2016; Lample et al., 2016), which combines bidirectional LSTM (Graves and Schmidhuber, 2005) with conditional random field (Lafferty, McCallum, and Pereira, 2001). The second model's sequence tagger is based on BERT (Devlin et al., 2019), which achieved state-of-the-art result for the CoNLL dataset.

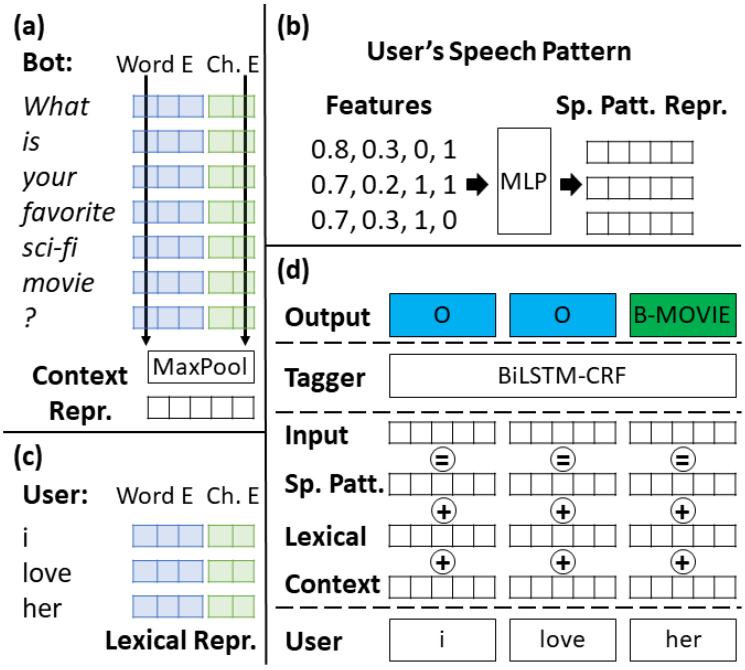

FIGURE 2.2: Models' structure. (a) Aggregate context using bag of embeddings. (b) Construct lexical representations of tokens in user's utterance. (c) Construct representations from speech pattern features. (d) Combine context, lexical, and speech pattern representations, and then output the tokens' tags. Word E: word embedding, Ch. E: character embedding, SP: speech pattern

Figure 2.2 shows the models' structure. The context encoder is a bag-of-embedding model (Figure 2.2a), which encodes the bot's utterance and outputs a single context vector. Specifically, the tokens' embeddings (concatenation of word and character embeddings) in the bot's utterance

are fed through a max-pooling layer to produce the context vector. The context vector and the lexical vectors (Figure 2.2b) are combined as models' input using element-wise addition (Figure 2.2d). The speech pattern encoder is a BiLSTM (Figure 2.2c), which encodes speech pattern features as vectors. These vectors are concatenated with the outputs from the last hidden layer of BiLSTM or BERT. While BiLSTM uses a conditional random field to tag the tokens, BERT uses a fully-connected layer instead (similar to (Devlin et al., 2019)).

Since BERT uses sub-word tokens, some words may be split into multiple tokens. For example, "*interstellar*" is split into "*inter*" and "*#stellar*". However, as the speech pattern features are only available for individual words and not for word pieces, these features have to be split up for multi-token words. In particular, the sub-word tokens have the same ASR confidence and duration as the word's ASR confidence and duration. Although the durations of the sub-word tokens should be shorter than the word's duration, it is not clear how to derive the correct durations. For the pauses, the preceding pause value is assigned to the first sub-word token while the succeeding pause value is assigned to the last sub-word token. Similar to (Devlin et al., 2019), a fully-connected layer is used to predict the tags instead of using conditional random field, and only the first sub-word token of the words are used to predict the tags.

3 Experiments

3.1 Data

The data are from conversations between humans and the Gunrock chatbot (Liang et al., 2020), which participated in the 2019 Amazon Alexa Prize. Conversations were collected during the period from December 2019 to May 2020. Each data sample consists of one chatbot utterance and the following human utterance (Figure 2.1). Chatbot utterances are in mixed-case while human utterances are output from an ASR system and are in lower case.

		Tokens		Avg. Len.	
	Turns	Bot	User	Bot	User
Train	22,908	624,168	146,858	27.2	6.4
Standard Split					
Dev	3,000	80,749	19,585	26.9	6.5
Test	3,000	81,668	19,279	27.2	6.4
Hard Split					
Dev	3,000	81,585	19,984	27.1	6.6
Test	3,000	82,137	20,583	27.3	6.8

TABLE 3.1: Data statistics. The data were collected during the period from December 2019 to May 2020. The data are divided into two different splits (standard and hard) with a shared training set. The hard split is used to test the robustness of the proposed model while the standard split is common practice in machine learning.

The data are divided into two different splits: a standard split and a hard split, and the two splits share the same training set (Table 3.1). While the training, development, and test set of the standard split are formed by randomly partitioning the data, the development and test set of the

	Standard Split	Hard Split
Dev	46.26%	14.45%
Test	46.75%	14.36%

TABLE 3.2: Number of unique named entities that are also in the training set (vocabulary transfer)

hard split are created such that they have more named entities that are not seen in the training set (i.e. little named entity overlap). Table 3.2 illustrates the difference in term of named entity overlap measured using vocabulary transfer rate (Palmer and Day, 1997). Vocabulary transfer is the proportion of unique named entities appearing in both training and test set, and as expected, the development and test sets of the hard split have much lower vocabulary transfer than that of the standard split. Although standard split is a common practice in machine learning, deep learning models can perform well on the standard split by exploiting the spurious patterns in the data (Jia and Liang, 2017). Thus, the hard split is necessary for measuring how well the models can generalize, since NER models relying heavily on surface patterns will underperform when there are a lot of unseen named entities (Augenstein, Derczynski, and Bontcheva, 2017). Furthermore, the test set of the hard split more closely resembles the test data during deployment because the data the models see during deployment usually differ from the data collected during training (little overlap of named entities). Thus, the performance on the hard split is a more realistic reflection of the models performance during deployment. A comparison between the size of the dataset used in this paper and that of popular public NER datasets is shown in Table 3.3.

Although named entities are typically classified into three big types: *Person*, *Location*, and *Organization* (Nadeau and Sekine, 2007), fine-grained typing may be more useful, especially for question-answering and information retrieval (Fleischman, 2001). For example, *Location* can be subdivided into *City*, *State*, and *Country* (Lee and Lee, 2005). Similarly, *Person* can be subdivided into *Politician* and *Entertainer* (Fleischman and Hovy, 2002). In addition, special types may be used to address systems' specific needs, for example *Film* (Etzioni et al., 2005), *Book title* (Brin, 1998;

	Train	Dev	Test
Number of Tokens			
CoNLL	203,621	51,362	46,435
OntoNotes	1,088,503	147,724	152,728
WNUT	62,730	15,733	23,394
Standard split	146,858	19,585	19,279
Hard split	146,858	19,984	20,583
Number of Entities			
CoNLL	23,499	5,942	5,648
OntoNotes	81,829	11,066	11,257
WNUT	1,975	836	1,079
Standard split	7,402	934	952
Hard split	7,402	1,254	1,391

TABLE 3.3: Comparing the dataset used in this paper against public NER datasets.

Witten et al., 1999), *Brand* (Bick, 2004), *Protein* (Shen et al., 2003; Tsuruoka and Tsujii, 2003; Settles, 2004), *Drug* (Rindflesch et al., 1999), and *Chemical* (Narayanaswamy, Ravikumar, and Vijay-Shanker, 2002).

Since the Gunrock chatbot needs to converse with users in different topics, fine-grained typing is more useful for accurately retrieving information about named entities. Named entities in data samples were manually labelled by Gunrock team members using 6 named entity types: *Movie*, *Book*, *Song*, *Person*, *Character*, and *Other*. The BIO scheme was used for labeling the data. Figure 3.1 and Table 3.4 show the distribution of named entities by types and the average entity length by types respectively. The *Movie*, *Book*, and *Song* types are for names of movies and TV shows, books, and songs respectively. The *Person* type includes names of real people or musical groups (e.g. Tom Hanks or Imagine Dragons). The *Character* type includes names of fictional people in movies or stories (e.g. Anna and Elsa in the movie Frozen). The *Other* type is for the other named entities (e.g. US or Siri) that do not belong to any of the previous 5 types. For labeling polysemous entities, context (i.e. chatbot utterance) is taken into account to determine the correct type. For example, for the human response "*yes harry potter*", "*harry potter*" is a *Character* with regard to the question

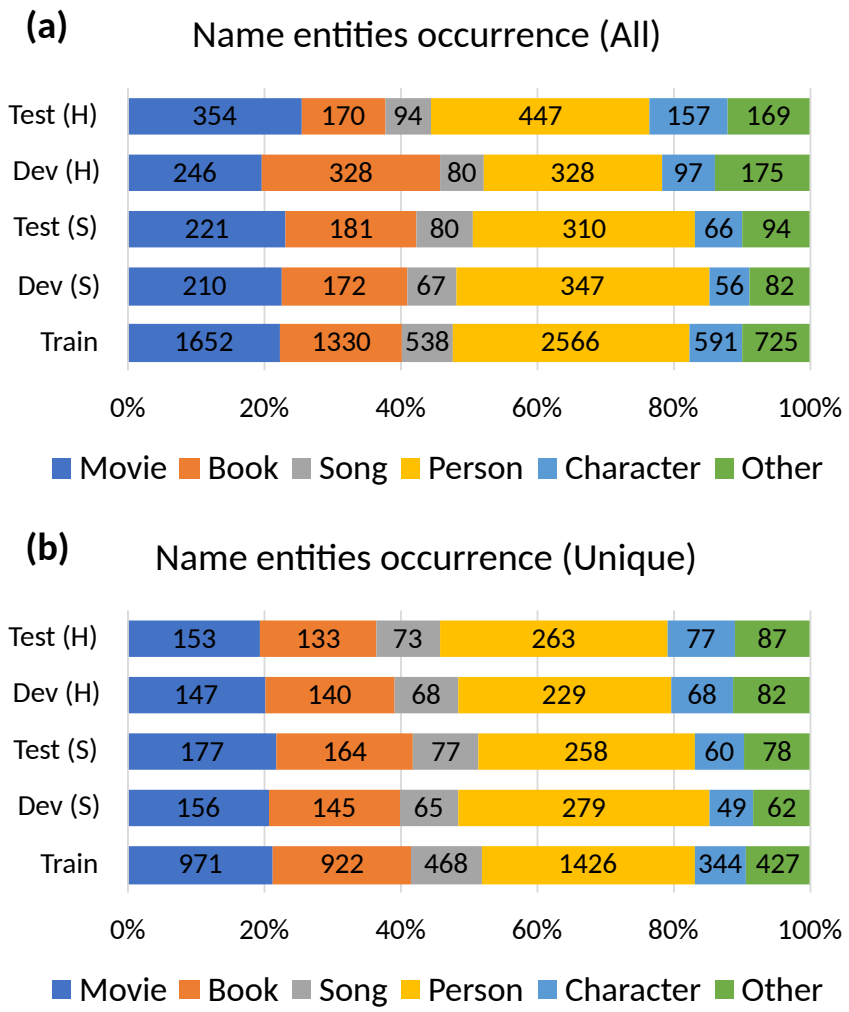

FIGURE 3.1: Entities by types, S: Standard, H: Hard

"Do you have a favorite character in the book?". However, when the question is *"Did you watch any movie recently?"*, *"harry potter"* is labeled as a *Movie*.

3.2 Implementation Details

The models are implemented using PyTorch (Paszke et al., 2019) and *transformers* (Wolf et al., 2020) libraries. For BiLSTM-CRF models, word embeddings and character embeddings were concatenated to form the context input and lexical input. The size of word embeddings and character

Type	*Movie*	*Book*	*Song*
Average Length	2.3	3.0	2.7
Type	*Person*	*Character*	*Other*
Average Length	2.0	1.3	1.6

TABLE 3.4: Average entity length (tokens) by entity types

embeddings are 300 and 100 respectively. Word embeddings were initialized using GloVe word vectors from (Pennington, Socher, and Manning, 2014). For BERT models, lexical input only includes sub-word embeddings. The size of the context encoder's word embedding and character layer are 600 and 168 respectively (so that the concatenated size is 768, matching the dimension of BERT). The parameters of the BERT model were initialized using the pre-trained uncased BERT base model. The speech pattern encoder is a two-layer BiLSTM with the hidden state size of 256. The dropout (Srivastava et al., 2014) rate of the speech pattern encoder was set at 0.3. The input to the encoder are speech pattern features which include: token ASR confidence, token duration, the pauses preceding and succeeding the token. Due to constraints in the Alexa data collection, other acoustic/prosodic speech features are unavailable. The token duration is thresholded at 1.5 second which is the 99th percentile value. The preceding (succeeding) pause is a binary variable, indicating whether there is a gap more than 30 milliseconds before (after) the token.

BiLSTM-CRF	
Learning rate	3e-3, 1e-3, 3e-4, 1e-4, 3e-5
Dropout	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
Dimension	128, 256, 512
BiLSTM layers	1, 2, 3, 4
Weight decay	1e-7, 1e-6, 1e-5
BERT	
Learning rate	1e-4, 6e-5, 3e-5, 1e-5
Weight decay	0.01

TABLE 3.5: Hyperparameter grids for random search

All models were trained for 100 epochs with the batch size of 128. BiLSTM-CRF models were trained using Adam (Kingma and Ba, 2014), while BERT models were trained using AdamW (Loshchilov

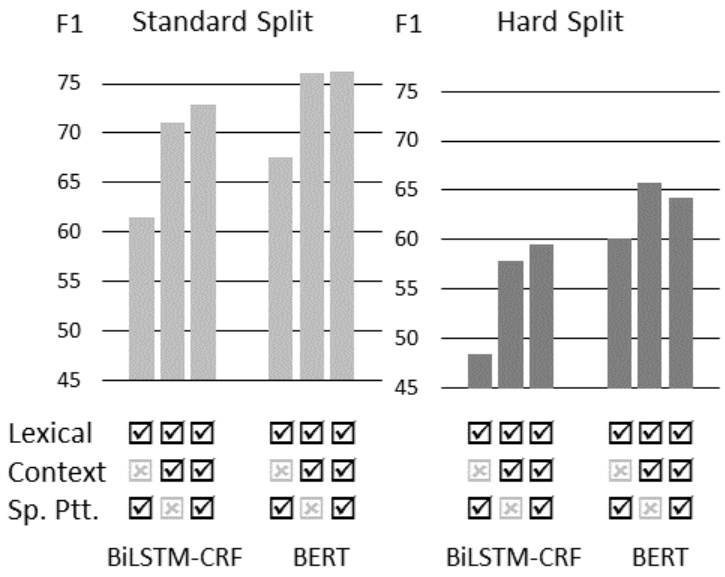

FIGURE 3.2: Context is always beneficial while speech pattern features are more beneficial in the hard split evaluation. Detailed results are in Table 3.6.

and Hutter, 2018). Linear learning rate schedule is used for training BERT whereby learning rate peaks after 10% of the training steps and then decreases to 0. We find models' hyperparameters using random search (Bergstra and Bengio, 2012) in 80 trials (see Table 3.5).

3.3 Results

Following CoNLL evaluation method, the models are evaluated using F1 score computed using complete spans of named entities. As shown in Figure 3.2, modeling context consistently leads to significant gain in F1 score, regardless of the data split or the model structure. For the standard split, the BiLSTM-CRF's F1 improved from 62.8% to 70.8% while BERT's F1 improved from 67.3% to 72.4%. Similarly for the hard split, the BiLSTM-CRF's F1 improved from 48.0% to 56.1% while BERT's F1 improved from 59.2% to 64.7%.

Adding speech pattern features did not lead to notable changes in F1 score when testing on the standard split. BiLSTM-CRF's F1 improved by 0.2% (62.8% to 63.0%) while BERT's F1 improved by 0.6% (67.3% to 67.9%) (see Table 3.6). However, when testing on the hard split, the gap between

Standard Split						
	Lx.	Ct.	SP	P	R	F1
LSTM	Y			59.8	66.1	62.8
LSTM	Y	Y		69.2	72.4	70.8
LSTM	Y		Y	58.3	68.6	63.0
LSTM	Y	Y	Y	69.5	73.2	**71.3**
BERT	Y			66.4	68.3	67.3
BERT	Y	Y		71.1	73.7	72.4
BERT	Y		Y	65.2	70.9	67.9
BERT	Y	Y	Y	71.1	75.1	**73.0**
Hard Split						
	Lx.	Ct.	SP	P	R	F1
LSTM	Y			42.5	55.1	48.0
LSTM	Y	Y		51.3	62.0	56.1
LSTM	Y		Y	42.6	57.6	49.0
LSTM	Y	Y	Y	51.8	65.6	**57.9**
BERT	Y			56.0	62.8	59.2
BERT	Y	Y		62.9	66.7	64.7
BERT	Y		Y	55.6	65.3	60.1
BERT	Y	Y	Y	62.5	69.0	**65.6**

TABLE 3.6: Context and speech pattern features improve NER performance. Lx.: Lexical, Ct.: Context, SP: Speech pattern features

using and not using speech pattern features is more noticeable. BiLSTM-CRF's F1 improved by 1.0% (48.0% to 49.0%) while BERT's F1 improved by 0.9% (59.2% to 60.1%). This is perhaps unsurprising since the lexical overlap (i.e. number of shared named entities) between the standard split's training and test set is quite high (see Table 3.2), so exploiting complementary features like speech pattern may be less beneficial.

In all setups, combining speech pattern features with context resulted in the highest F1 scores. Besides, BERT models outperformed BiLSTM-CRF models as the former were pre-trained on a large amount of data while the latter were trained from scratch. Lastly, performance on the hard split is still lower than that on the standard split, indicating room for improving the models' robustness.

Standard Split						
	Lx.	Ct.	SP	P	R	F1
BERT 4F	Y		Y	65.2	70.9	67.9
BERT 3F	Y		Y	65.9	68.7	67.2
BERT 2F	Y		Y	66.0	70.2	**68.0**
BERT 4F	Y	Y	Y	71.1	75.1	73.0
BERT 3F	Y	Y	Y	71.7	76.2	73.9
BERT 2F	Y	Y	Y	72.2	77.7	**74.8**
Hard Split						
	Lx.	Ct.	SP	P	R	F1
BERT 4F	Y		Y	55.6	65.3	**60.1**
BERT 3F	Y		Y	56.8	62.9	59.7
BERT 2F	Y		Y	55.5	62.2	58.7
BERT 4F	Y	Y	Y	62.5	69.0	**65.6**
BERT 3F	Y	Y	Y	62.3	66.9	64.5
BERT 2F	Y	Y	Y	60.6	67.1	63.7

TABLE 3.7: Speech pattern features ablation. 4F: all features, 3F: without ASR confidence, 2F: without ASR confidence and token duration. Lx.: Lexical, Ct.: Context, SP: Speech pattern features

3.4 Ablation

In order to determine the usefulness of different speech pattern features, we conducted ablation study by removing the features one by one. In particular, starting with a model that uses all 4 features (denoted as 4F): namely token ASR confidence, token duration, the pauses preceding and succeeding the token, we first remove the ASR confidence from the model input (denoted as 3F) and then remove the token duration from the model input (denoted as 2F). We trained all the models with ablated features from scratch with hyperparameter search similar to what was done in Section 3.2.

For the hard split, the BERT 4F model did better than the BERT 3F model, showing that the ASR confidence is probably useful. Low ASR confidence can indicate names which appear infrequently (e.g. ASR: "herman hess", ASR confidence [0.3, 0.1], actual name: "Hermann Hesse"). Similarly, the BERT 3F model did better than the BERT 2F model, suggesting that token duration is also

Standard Split						
	Lx.	Ct.	SP	P	R	F1
LSTM	Y	Y	Y	69.5	73.2	71.3
BERT†	Y	Y	Y	62.9	70.6	66.5
BERT	Y	Y	Y	71.1	75.1	**73.0**
Hard Split						
	Lx.	Ct.	SP	P	R	F1
LSTM	Y	Y	Y	51.8	65.6	57.9
BERT†	Y	Y	Y	41.4	55.9	47.5
BERT	Y	Y	Y	62.5	69.0	**65.6**

TABLE 3.8: Effect of pre-training. Lx.: Lexical, Ct.: Context, SP: Speech pattern, †: trained from scratch

probably useful. Surprisingly, for the standard split BERT 2F outperformed BERT 4F, suggesting that ASR confidence and token duration may be less useful when there is high lexical overlap.

Although, the pre-trained BERT model beat the BiLSTM-CRF model (Section 3.3), when the BERT model is trained from scratch, it did worse than the BiLSTM-CRF model (Table 3.8). Evidently, pre-training provided a massive boost in performance. Although, the NER performance of BERT training from scratch could be improved via extensive hyperparameter search, BiLSTM-CRF is a competitive model when pre-training is not viable.

4 Discussion

4.1 Roles of context and speech patterns

Although unknown words may pose a challenge to NER systems, entities that have multiple types are harder to deal with than unknown words (Bernier-Colborne and Langlais, 2020). Dialog context may help resolving the type of an entity when the entity belongs to multiple types. Figure 4.1 [1] shows that, without context, both BiLSTM-CRF and BERT predicted "lord of the rings" as *Book* (incorrect) instead of *Movie*. Knowing dialog context also helps when named entities are common phrases. Without context, BiLSTM-CRF missed the entity "the notebook", while BERT misclassified it as *Book*.

Bot	*Do you have a favorite fantasy movie ?*
User	lord of the rings
LSTM w/o context	**[lord of the rings]Book**
LSTM with context	[lord of the rings]Movie
BERT w/o context	**[lord of the rings]Book**
BERT with context	[lord of the rings]Movie

Bot	*What movie would you recommend ?*
User	i would recommend the notebook
LSTM w/o context	—
LSTM with context	[the notebook]Movie
BERT w/o context	**[the notebook]Book**
BERT with context	[the notebook]Movie

FIGURE 4.1: Without context, both models either predicted the wrong entity type or missed the named entity.

[1] Examples shown in this section are from internal user studies and are not in the training, development, or test sets. Users have given consent for the release of these examples. Some parts have been anonymized to protect users' privacy.

In contrast, speech pattern features may help locating the named entities. Figure 4.2 shows that NER models without speech pattern features might predict the wrong text spans as named entities (e.g. "*jonas brothers once*" instead of "*jonas brothers*"). Interestingly, although the predicted

Bot	*Have you been to a live performance ?*
User	yes i saw the jonas brothers once
Pauses	yes i saw the jonas brothers once
Confidence	0.9, 0.9, 0.9, 0.9, 0.9, 0.9, **0.8**
LSTM w/o SP	**[jonas brothers once]Person**
LSTM with SP	[jonas brothers]Person
BERT w/o SP	**[jonas brothers once]Person**
BERT with SP	[the jonas brothers]Person

Bot	*What's the last movie that made you laugh ?*
User	i'm not sure probably the movie with mclovin
Pauses	i'm not sure **PAUSE** probably **PAUSE** the movie **PAUSE** with **PAUSE** mclovin
Confidence	0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, **0.0**
LSTM w/o SP	**[with mclovin]Movie**
LSTM with SP	[mclovin]Movie
BERT w/o SP	**[mclovin]Person**
BERT with SP	[mclovin]Person

FIGURE 4.2: Speech pattern helps locating named entities. Without speech pattern, models predicted the wrong entity spans (e.g. "*jonas brothers once*" and "*with mclovin*"). SP: speech patterns

type is not correct, the type of "*mclovin*" predicted by BERT is more plausible than BiLSTM-CRF. This might be because BERT gained some world knowledge after pre-training, and NER models usually benefit from external sources of knowledge (Ratinov and Roth, 2009; Passos, Kumar, and McCallum, 2014).

4.2 Towards robust NER in dialog system

Current ASR systems still perform poorly in domains that require special vocabulary and under noisy conditions (Georgila et al., 2020). Unfamiliar words or recording noise may lead to ASR errors that affect downstream tasks such as NER. Although continuously retraining the ASR and

NER models can reduce these errors, such effort may be costly. Integrating features such as speech pattern features, which are less affected by changing vocabulary and recording conditions, could make NER models more robust and reduce the frequency of having to retrain the models.

Speech pattern features have been used for NER in spoken broadcast news although this did not lead to improvement in performance (Hakkani-Tür et al., 1999). This could be because these features might also encode other phenomena such as stressing that are not relevant for NER task (Hakkani-Tür et al., 1999). In contrast to (Hakkani-Tür et al., 1999) where the features encoder and the NER tagging model were trained, we trained the models jointly so they are more sensitive to cases when speech pattern features are indicative of named entities. Our proposed models show consistent improvement over lexical-features-only baselines, especially when training and testing data are significantly different, demonstrating that it is possible to combine lexical and speech pattern features to achieve more robust NER system.

4.3 Future work

We show that short context and minimal speech pattern features can improve NER performance. Better performance might be achieved by modeling longer context and more features (e.g. prosodies, parts of speech, punctuation) from a state-of-the-art ASR system. Prosodic features can also be extracted automatically to better align to sub-word tokens (Tran et al., 2018). It would also be interesting to see how robust NER would improve entity linking especially when entity mentions contain ASR errors.

Since our work only explored open-domain conversations between humans and a chatbot, it is important to validate the benefits of modeling context and speech pattern features in other settings. Examples of other settings include open-domain conversations between humans or task-oriented conversations between humans or between humans and chatbots. For these different settings, NER models might need longer context or speech pattern features other than what were used in this paper. However, many previous studies have shown the usefulness of these additional

features in other tasks so there are reasons to believe that the findings should translate to other datasets and settings.

4.4 Conclusions

Named entity recognition for dialogs is difficult because utterances are ambiguous out of context and ASR transcripts are noisy due to ASR errors and the lack of punctuation and capitalization. We proposed two NER models exploiting dialog context and speech patterns to address the ambiguity issue and ASR noise. Our results show that context usually improves NER accuracy while speech patterns help in the more difficult but more realistic scenario with many unseen named entities. Further studies on exploiting features from non-text modalities are warranted to enhance NER in dialog systems.

Bibliography

Akbik, Alan, Tanja Bergmann, and Roland Vollgraf (2019). "Pooled contextualized embeddings for named entity recognition". In: *Proceedings of NAACL*, pp. 724–728. DOI: 10.18653/v1/N19-1078.

Augenstein, Isabelle, Leon Derczynski, and Kalina Bontcheva (2017). "Generalisation in named entity recognition: A quantitative analysis". In: *Computer Speech & Language* 44, pp. 61–83.

Béchet, Frédéric et al. (2004). "Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How May I Help You? sm, tm". In: *Speech Communication* 42.2, pp. 207–225.

Bergstra, James and Yoshua Bengio (2012). "Random search for hyper-parameter optimization". In: *The Journal of Machine Learning Research* 13.1, pp. 281–305.

Bernier-Colborne, Gabriel and Philippe Langlais (2020). "HardEval: Focusing on Challenging Tokens to Assess Robustness of NER". In: *Proceedings of LREC*, pp. 1704–1711.

Bick, Eckhard (May 2004). "A Named Entity Recognizer for Danish". In: *Proceedings of LREC*. Lisbon, Portugal: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2004/pdf/99.pdf.

Brin, Sergey (1998). "Extracting patterns and relations from the world wide web". In: *International Workshop on The World Wide Web and Databases*. Springer, pp. 172–183.

Caubrière, Antoine et al. (2020). "Where are we in Named Entity Recognition from Speech?" In: *Proceedings of LREC*, pp. 4514–4520.

Cervantes, Gerardo and Nigel Ward (2020). "Using Prosody to Spot Location Mentions". In: *Proceedings of Speech Prosody 2020*, pp. 915–919. DOI: 10.21437/SpeechProsody.2020-187. URL: http://dx.doi.org/10.21437/SpeechProsody.2020-187.

Chen, C Julian (1999). "Speech recognition with automatic punctuation". In: *Sixth European Conference on Speech Communication and Technology*.

Derczynski, Leon et al. (2017). "Results of the WNUT2017 shared task on novel and emerging entity recognition". In: *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 140–147.

Devlin, Jacob et al. (2019). "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *Proceedings of NAACL*. DOI: 10.18653/v1/N19-1423.

Etzioni, Oren et al. (2005). "Unsupervised named-entity extraction from the web: An experimental study". In: *Artificial intelligence* 165.1, pp. 91–134.

Fitt, Susan (1995). "The pronunciation of unfamiliar native and non-native town names". In: *Proceedings of European Conference on Speech Communication and Technology*.

Fleischman, Michael (2001). "Automated subcategorization of named entities". In: *ACL (Companion Volume)*, pp. 25–30.

Fleischman, Michael and Eduard Hovy (2002). "Fine grained classification of named entities". In: *Proceedings of COLING*.

Galibert, Olivier et al. (2014). "The ETAPE speech processing evaluation". In: *Proceedings of LREC*, pp. 3995–3999.

Georgila, Kallirroi et al. (2020). "Evaluation of Off-the-shelf Speech Recognizers Across Diverse Dialogue Domains". In: *Proceedings of LREC*, pp. 6469–6476.

Goldman-Eisler, Frieda (1958). "Speech production and the predictability of words in context". In: *Quarterly Journal of Experimental Psychology* 10.2, pp. 96–106.

Graves, Alex and Jürgen Schmidhuber (2005). "Framewise phoneme classification with bidirectional LSTM and other neural network architectures". In: *Neural networks* 18.5-6, pp. 602–610.

Grishman, Ralph and Beth M Sundheim (1996). "Message understanding conference-6: A brief history". In: *Proceedings of COLING*.

Hakkani-Tür, Dilek et al. (1999). "Combining words and prosody for information extraction from speech". In: *Sixth European Conference on Speech Communication and Technology*.

Horlock, James and Simon King (2003). "Discriminative Methods for Improving Named Entity Extraction on Speech Data". In: *Eighth European Conference on Speech Communication and Technology*.

Hovy, Eduard et al. (2006). "OntoNotes: the 90% solution". In: *Proceedings of NAACL*, pp. 57–60.

Jia, Robin and Percy Liang (2017). "Adversarial examples for evaluating reading comprehension systems". In: *Proceedings of EMNLP*. DOI: 10.18653/v1/D17-1215.

Jurafsky, D. and J.H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall.

Katerenchuk, Denys and Andrew Rosenberg (2014). "Improving named entity recognition with prosodic features". In: *Fifteenth Annual Conference of the International Speech Communication Association*.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *Proceedings of ICLR*.

Kubala, Francis et al. (1998). "Named entity extraction from speech". In: *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pp. 287–292.

Lafferty, John D, Andrew McCallum, and Fernando CN Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of ICML*, pp. 282–289.

Lample, Guillaume et al. (2016). "Neural Architectures for Named Entity Recognition". In: *Proceedings of NAACL*, pp. 260–270.

Lee, Seungwoo and Gary Geunbae Lee (2005). "Heuristic methods for reducing errors of geographic named entities learned by bootstrapping". In: *Proceedings of IJCNLP*. Springer, pp. 658–669.

Lenzi, Valentina Bartalesi, Manuela Speranza, and Rachele Sprugnoli (2012). "Named entity recognition on transcribed broadcast news at EVALITA 2011". In: *International Workshop on Evaluation of Natural Language and Speech Tool for Italian*. Springer, pp. 86–97.

Liang, Kaihui et al. (2020). "Gunrock 2.0: A user adaptive social conversational system". In: *Proceedings of the 3rd Alexa Prize (Alexa Prize 2020)*.

Loshchilov, Ilya and Frank Hutter (2018). "Decoupled Weight Decay Regularization". In: *Proceedings of ICLR*.

Ma, Xuezhe and Eduard Hovy (2016). "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: *Proceedings of ACL*, pp. 1064–1074.

Mayhew, Stephen, Nitish Gupta, and Dan Roth (2020). "Robust Named Entity Recognition with Truecasing Pretraining". In: *Proceedings of AAAI*.

Nadeau, David and Satoshi Sekine (2007). "A survey of named entity recognition and classification". In: *Lingvisticae Investigationes* 30.1, pp. 3–26.

Narayanaswamy, Meenakshi, KE Ravikumar, and K Vijay-Shanker (2002). "A biological named entity recognizer". In: *Biocomputing 2003*. World Scientific, pp. 427–438.

Palmer, David D and David Day (1997). "A statistical profile of the named entity task". In: *Fifth Conference on Applied Natural Language Processing*, pp. 190–193.

Palmer, David D and Mari Ostendorf (2001). "Improving information extraction by modeling errors in speech recognizer output". In: *Proceedings of the first international conference on Human language technology research*.

Passos, Alexandre, Vineet Kumar, and Andrew McCallum (2014). "Lexicon Infused Phrase Embeddings for Named Entity Resolution". In: *Proceedings of CoNLL*, pp. 78–86.

Paszke, Adam et al. (2019). "Pytorch: An imperative style, high-performance deep learning library". In: *Proceedings of NeurIPS*, pp. 8026–8037.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of EMNLP*, pp. 1532–1543.

Rangarajan, Vivek and Shrikanth Narayanan (2006). "Detection of non-native named entities using prosodic features for improved speech recognition and translation". In: *Multilingual Speech and Language Processing*.

Ratinov, Lev and Dan Roth (2009). "Design challenges and misconceptions in named entity recognition". In: *Proceedings of CoNLL*, pp. 147–155.

Reich, Shuli S (1980). "Significance of pauses for speech perception". In: *Journal of Psycholinguistic Research* 9.4, pp. 379–389.

Rindflesch, Thomas C et al. (1999). "EDGAR: extraction of drugs, genes and relations from the biomedical literature". In: *Biocomputing 2000*. World Scientific, pp. 517–528.

Settles, Burr (2004). "Biomedical named entity recognition using conditional random fields and rich feature sets". In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pp. 107–110.

Shen, Dan et al. (2003). "Effective adaptation of hidden markov model-based named entity recognizer for biomedical domain". In: *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, pp. 49–56.

Shriberg, Elizabeth et al. (2000). "Prosody-based automatic segmentation of speech into sentences and topics". In: *Speech communication* 32.1-2, pp. 127–154.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *JMLR* 15.

Sudoh, Katsuhito, Hajime Tsukada, and Hideki Isozaki (2006). "Incorporating speech recognition confidence into discriminative named entity recognition of speech data". In: *Proceedings of ACL*, pp. 617–624.

Sundheim, Beth M (1995). "OVERVIEW OF RESULTS OF THE MUC-6 EVALUATION". In: *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Tjong Kim Sang, Erik F and Fien De Meulder (2003). "Introduction to the CoNLL-2003 shared task: language-independent named entity recognition". In: *Proceedings of NAACL*, pp. 142–147.

Tran, Trang et al. (2018). "Parsing Speech: A Neural Approach to Integrating Lexical and Acoustic-Prosodic Information". In: *Proceedings of NAACL*, pp. 69–81. DOI: 10.18653/v1/N18-1007.

Tsuruoka, Yoshimasa and Jun'ichi Tsujii (2003). "Boosting precision and recall of dictionary-based protein name recognition". In: *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, pp. 41–48.

Turmo, Jordi et al. (2009). "Overview of QAST 2009". In: *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, pp. 197–211.

Witten, Ian H et al. (1999). "Using language models for generic entity extraction". In: *Proceedings of the ICML Workshop on Text Mining*. Citeseer, p. 14.

Wolf, Thomas et al. (2020). "Transformers: State-of-the-art natural language processing". In: *Proceedings of EMNLP: System Demonstrations*, pp. 38–45.