Turn Detection In Textual Conversations

امین پوردبیری
دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
amin.pourdabiri@gmail.com

سیده زهرا سیدی
دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
s_sayedi@comp.iust.ac.ir

چکیده

مساله تشخیص تغییر گوینده[۱]، تعدادی مکالمه که به صورت مرتب وجود دارند را به عنوان ورودی دریافت کرده و تشخیص می‌دهد چه موقعی گوینده آن مکالمه تغییر خواهد کرد. در این مساله هر مکالمه با برچسب‌های صفر و یک شناخته می‌شود. مجموعه دادگان مربوط به مکالمات برای آموزش دادن عامل‌های مکالمه هوشمند به کار می‌رود. هرچند بسیاری از این مجموعه دادگان همانند دیتاست OpenSubtitle فاقد برچسبی از این جنس می‌باشد، یعنی در آن تغییر گوینده‌ها لیبل زده نشده است. مقالات گذشته[۹] برای استفاده از این مجموعه دادگان راه‌های ساده‌ای را پیش گرفته بودند. آن‌ها به این روش عمل می‌کردند که فرض می‌شد هر مکالمه مختص یک فرد جدید می‌باشد. در این پروژه ما تعدادی رده‌بند[۲] را آموزش می‌دهیم که با کمک ویژگی‌های استخراج شده از مجموعه دادگان SwDA[۳] این کار ممکن خواهد بود. وقتی مدل‌ها را بر روی این دیتاست آموزش دادیم، می‌توانیم دقت مدل‌ها را با استفاده از تعدادی مکالمه که به صورت دستی برای مجموعه دادگان OpenSubtitle[۴] لیبل زده‌ایم، به دست آوریم. در نتیجه این مدل می‌تواند برای برچسب زدن خودکار داده‌های مجموعه دادگان OpenSubtitle مورد استفاده قرار گیرد و کیفیت عامل‌های مکالمه هوشمند[۵] را بهبود بخشد.

۱ مقدمه

یکی از نکات مهم در طراحی یک عامل مکالمه هوشمند استفاده از مجموعه دادگان مناسب و کامل می‌باشد. یکی از دیتاست‌هایی که بسیار مورد استفاده قرار می‌گیرد OpenSubtitle نام دارد.

این دیتاست بسیار ارزشمند در زمینه دیالوگ می‌باشد در صورتی که لیبل گوینده ندارد. پس عامل مکالمه آموزش داده شده بر روی این دیتاست نمی‌تواند متوجه شود کدام مکالمه در جواب یک مکالمه دیگر آمده است. به عبارت دیگر مکالمات متوالی در این زیر نویس‌ها به گونه‌ای ذخیره شده است که مشخص نیست کدام مکالمه در ادامه دیگری قرار می‌گیرد. برای استفاده از این دیتاست در عامل مکالمه باید به روشی این لیبل‌های تغییر گوینده را به دست آوریم. اما همان‌طور که گفت شد، تاکنون عامل‌های مکالمه هوشمندی که بر روی این دیتاست ساخته شده اند، نمی‌توانند چنین جملاتی را به درستی مورد استفاده قرار دهند زیرا هر دو مختص یک گوینده می‌باشند:

- من می‌خواهم به خانه بروم
- اما نمی‌دانم اکنون کجا هستم

[1] Turn detection
[2] Classifer
[3] https://github.com/cgpotts/swda
[4] http://opus.nlpl.eu/OpenSubtitles.php
[5] Intelligent dialog system

گوینده	جمله	لیبل
A	Uh, taxes	New
B	Do you want to go ahead?	New
B	Sure	Continue
B	We pay far too much in taxes	Continue
B	Well, far too much for what we get	Continue
B	I mean	Continue
A	Uh,huh	New
B	It's just	New
A	I agree	New
B	I don't know. It ju.., just seems	New
	too much of the money is just lost	

جدول ۱: مثالی از مکالمه‌ای که دارای لیبل گوینده می‌باشد و هر جمله نیز با لیبل مختص خود به نمایش درآمده است

در کل هدف این پروژه شناسایی تغییر گوینده در مکالمات متنی است، به گونه‌ای که گوینده جمله و کسی که به آن پاسخ می‌دهد از یکدیگر قابل تمایز باشند. اما نکته‌ای که این‌جا مطرح می‌باشد، وقت‌گیر و هزینه‌بر بودن لیبل گذاری مجموعه دادگان OpenSubtitle است. به همین دلیل ما از دیتاست دارای لیبل SwDA که به ما در این فرایند کمک خواهد کرد استفاده خواهیم کرد تا بتوانیم OpenSubtitle را لیبل گذاری نماییم.

۲ کارهای مرتبط / پیش‌زمینه

در این بخش ما به تعدادی از کارهای مرتبط در این زمینه که در حوزه گفتار و یا مکالکمات متنی هستند، اشاره خواهیم کرد. یکی از این کارها استفاده از صوت و ویژگی‌های لغوی برای شناسایی تغییر گوینده استفاده کرده است[۵، ۴، ۶، ۳]. سه عدد از این مقالات در زبان ژاپنی می‌باشند[۴، ۶، ۳] در صورتی که یکی از آنها به زبان آلمانی پیاده سازی شده است[۵]. مقاله [۶] از شبکه ترتیبی stacked time-asynchronous برای شناسایی محل اتمام صحبت‌های یک گوینده کمک گرفته است. به این گونه که توالی‌ای از ویژگی‌های ناهمگام را بررسی می‌کند. مقاله [۳] ارتباط بین ویژگی‌های صرفی و نحوی را مورد بررسی قرار داده و نشان می‌دهد که ترکیب آن دو می‌تواند به این مساله کمک کند. مقاله [۴] از یک شبکه بازگشتی RNN برای تقسیم‌بندی مکالمات به ۴ کلاس کمک گرفته است که مرتبط با رفتار تغییر گوینده با استفاده از ترکیب ویژگی‌های صوتی و لغوی می‌باشد. در آخر مقاله [۵] با استفاده از یک شبکه LSTM همرا با یک دیکودر threshold-based و مورد بررسی قرار دادن موازنه بین تاخیر و نرخ cut-in برای شناسایی مکالمات به صورت real-time این کار را انجام داده است.

۳ مدل پیشنهاد شده

شناسایی تغییر گوینده یکی از مسائل پردازش زبان‌های طبیعی می‌باشد. این مساله کمک می‌کند که گوینده جمله و یا مکالمه را شناسایی کنیم. اهمیت این کار زمانی حس می‌شود که می‌خواهیم از مجموعه دادگانی استفاده کنیم که فاقد لیبل گوینده می‌باشد. دو نوع لیبل برای هر مکالمه قابل بیان است. یکی از آنها new و دیگری continue خواهد بود. اگر مکالمه کننده کنونی با قبلی متفاوت باشد به آن لیبل new و اگر گوینده جمله با جمله قبل یکسان باشد، continue اختصاص داده می‌شود. مثالی از این نمونه در جدول ۱ نمایش داده شده است.

۱.۳ دیتاست‌ها:

- SwDA [۲] مجموعه دادگانی متشکل از گفت و گوهای تلفنی که دارای ۱۱۲۶ مکالمه متفاوت می‌باشد و در سال ۱۹۹۳ منتشر شده است. همانطور که پیشتر اشاره شد، این دیتاست دارای تگ گوینده می‌باشد.
- OpenSubtitle مجموعه دادگانی متشکل از تعداد زیادی از زیرنویس فیلم ها که در حدود ۴ میلیون و ۸۰۰ هزار زیرنویس در ۶۲ زبان مختلف را دارا میباشد. از این مجموعه دادگان برای کاربرد های زیادی نظیر عامل‌های هوشمند استفاده شده است.

برای این پروژه باید قسمتی از دادگان آموزش یا همان تست را نیز در اختیار داشته باشیم. این مجموعه با لیبل زدن ۱۰۰۰ خط از opensubtitle فراهم شد. این کار امکان بررسی دقت مدل ساخته شده را ممکن می‌سازد.

۲.۳ انتخاب ویژگی‌ها:

انتخاب ویژگی‌ها یکی از مهم‌ترین بخش‌ها در زمینه کلاسیفیکیشن می‌باشد. زیرا انتخاب مناسب ویژگی این امکان را فراهم می‌سازد که مدل‌های بهتری با دقتی بیشتر آموزش دهیم و هم‌چنین در مقابل نویزها مقاومت بیشتری به وجود آوریم. دو الگو برای استخراج ویژگی‌ها می‌تواند مورد استفاده قرار گیرد:

۱. ویژگی‌های لغوی: تمامی کلمات همان‌طور که در عبارت ظاهر می‌شوند.
۲. ویژگی‌های POS: تمامی کلمات با نقششان در جمله جایگزین می‌شوند.
۳. ویژگی‌های مکانی: کلمات یا تگ‌های pos با توجه به نقششان اصلاح می‌شوند.
۴. ویژگی‌های خارج از جمله کنونی: کلمات یا تگ‌های pos جمله قبل

این ویژگی‌ها و روش‌های مختلف ترکیبشان در جدول ۲ آمده است.

نام	ویژگی‌ها	تعداد ویژگی‌ها
A	Unigram bag of words	۲۴/۵۳۵
B	Unigram words plus position (<pos>_<word>)	۱۱۶/۵۹۱
C	Numbers and punctuations removed, otherwise the same as A	۲۱/۳۰۶
D	Bigram bag of words	۲۳۶/۵۱۱
E	Unigram POS tag bag of words	۷۶
F	The same as E plus tag position (<pos>_<POSTag>)	۲/۵۲۳
G	The same as E while merging all tokens without a POS tag into a single feature	۶۹
H	Bigram POS tag bag of words	۱/۷۲۸
I	Unigram words plus POS tags bag of words (<word>_<POSTag>)	۳۳/۴۵۳
J	The same as "I" except for the first token of the utterance which also contains position (first_<firstword>_<POSTag>)	۳۷/۱۳۶
K	The same as "J" except for the last token of the utterance which also contains position information (last_<lastword>_<POSTag>)	۴۲/۲۰۰
L	The same as "K" plus adding the first and last token of the previous utterance. (prev_first_<firstword>, prev_last_<lastword>)	۶۳/۱۷۶
M	Features of "A" plus "E" plus first and last token of the utterance with position information and first and last token of the previous utterance without position information	۴۳/۶۱۹
N	Combining the features of "M" and "L"	۸۰/۹۵۳

جدول ۲: انواع ویژگی‌هایی که برای ساخت مدل استفاده شده اند

۳.۳ ابزار:

اولین ابزاری که برای رده‌بندی هر کدام از مکالمات استفاده کردیم Mallet [۷] است. این نرم افزار، ابزارهای پیچیده‌ای برای رده‌بندی در زمینه متن در اختیار ما قرار می‌دهد. روش‌های موثری برای تبدیل متن به ویژگی‌ها. الگوریتم‌های متنوع و زیادی نظیر naïve bayes ، maximum entropy ، درخت‌های تصمیم و کدهایی برای ارزیابی عملکرد این رده‌بندها به کمک روش‌هایی متنوع. هم‌چنین علاوه بر رده‌بند شامل ابزارهای دیگری نظیر NER در زمینه متن نیز می‌شود.

یکی دیگر از ابزارهای موجود در این نرم افزار sequence tagging می‌باشد. از آنجایی که دیتاست ما به صورت مکالمه‌ای است و ترتیب در آنها مهم می‌باشد، از این ویژگی نیز استفاده کردیم.

از دیگر ابزارهایی که برای ساختن مدل تشخیص turn استفاده کردیم شبکه‌های عصبی عمیق است. در این پروژه ما از شبکه‌های LSTM ، BERT ، BiLSTM و BiLSTM با لایه attention برای طبقه بندی متن کمک گرفتیم.

BERT (Bidirectional Encoder Representations from Transformers)[۱] یک شبکه پیش آموزش در زمینه متن است که روی دیتای ویکی‌پدیا و BookCorpus آموزش دیده است. این شبکه دو ورودی A و B می‌گیرد که A همان ورودی اصلی ما است و B برای مدل‌هایی استفاده می‌شود که ارتباط بین داده‌ها مهم باشد برای مثال در مساله ترجمه ماشینی در ورودی A جمله‌ای که باید ترجمه شود و در ورودی B ترجمه آن می‌آید یا به همین ترتیب برای مساله سوال و جواب. چون دیتای ما مکالمه است پس در قسمت B پاسخ هر قسمت باید باشد.

برای پیش پردازش باقی شبکه‌ها، پیش پردازش یکسان و به این صورت است: ۱- کوچک کردن حروف ۲- توکن بندی کردن ۳- تگ POS زدن ۴- تبدیل به وکتور کردن ۵- پد کردن

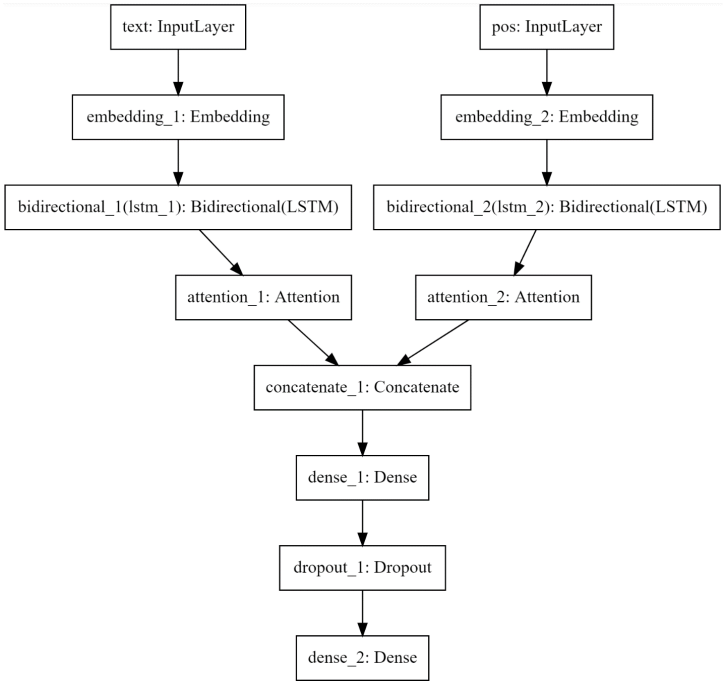
از آنجایی که مساله شناسایی گوینده یکی از مسائل پردازش متن می‌باشد، مرحله پیش پردازش آن را با سایر مسائل مانند sentiment analysis نباید یکسان دانست. در روش‌های رایج پیش پردازش، ما علائم و stop wordها را از متن حذف می‌کردیم. در صورتی که در اینجا می‌توانند بسیار به کمک ما آیند. برای مثال "؟" میتواند پایان یک جمله سوالی را برساند که به احتمال خیلی زیاد جمله بعد در جواب و پاسخ آن خواهد بود.

از نکات دیگر این است که تگ‌های POS می‌توانند کار ما را در مواردی بسیار راحت‌تر بکنند. برای مثال تمامی کلمات "من"، "تو"، "او" و... با عنوان PRP شناخته می‌شوند. کاری که ما در این شبکه انجام دادیم آزمایشات مختلفی در این زمینه بود. در مرحله اول فقط داده را به عنوان ورودی به شبکه دادیم. در مرحله دوم فقط تگ های POS و در آخرین مرحله به ترکیب آنها پرداختیم.

پس تا به اینجا ورودی‌های شبکه را تعریف کرده‌ایم. پس از این مرحله، باید لایه‌های شبکه را طراحی کنیم. انواع لایه‌های استفاده شده در این پروژه پیشتر ذکر شد. برای استفاده از آن‌ها ابتدا از word embedding ها استفاده کردیم. به این منظور باید این لایه با ابعاد مختلف در نظر گرفته شود. در این قسمت ما از GloVe کمک گرفتیم. این word embedding دارای نسخه‌هایی با ابعاد ۵۰، ۱۰۰، ۲۰۰ و ۳۰۰ می‌باشد. پس در وهله اول باید ببینیم کدام یک با مساله ما سازگاری بیشتری دارد. به طور میانگین سایز ۲۰۰ برای مجموعه دادگان ما بهترین نتیجه را از لحاظ بار زمانی و دقت حاصل می‌کرد. اما ما در نهایت سایز ۳۰۰ را به عنوان ورودی این لایه در نظر گرفتیم. اگرچه بار زمانی زیادی را متحمل می‌شود ولی دارای ویژگی‌های بیشتری است و به دلیل بزرگ بودن مجموعه دادگان ما کارایی بهتری از خود نشان می‌دهد.

برای تگ‌های POS نیز لایه embedding در نظر گرفتیم. این لایه را با ابعاد ۵۰ وارد شبکه کردیم و بعد از ورودی هاPOS قرار دادیم.

پس از این لایه باید به سراغ لایه‌های LSTM ، BiLSTM و Dense برویم. انواع ساختار این مدل‌ها در فایل ژوپیتر آپلود شده قابل مشاهده است. به همین منظور ما تنها ساختار کاملترین شبکه را در اینجا قرار می‌دهیم:



نکته قابل ذکر این است که برای لایه Attention نکته قابل ذکر این است که برای لایه Attention از این مقاله [۸] کمک گرفته‌ایم. در ابتدا این لایه را با زدن query بر روی جملات پرسشی پیاده سازی کردیم. یعنی با استفاده از لیستی که شامل این کلمات بود، آنها را شناسایی کرده و با افزون وزنشان مجددا به شبکه وارد می‌کردیم. اما از آنجا که با استفاده از لایه Attention پیاده سازی شده در مقاله مذکور به دقت بهتری رسیدیم، همین لایه را برای مدل خود مورد استفاده قرار دادیم.

در نهایت بعد از یکی کردن خروجی‌های لایه ،Attention آن را به لایه‌های Dense دادیم. از آنجا که خروجی ما ۲ کلاسه می‌باشد باید از activation ایfunction استفاده کنیم که یک عدد برگرداند. یکی از بهترین آنها sigmoid می‌باشد که عددی بین ۰ و ۱ به ما می‌دهد. در نهایت با گرد گردن هر کدام از آنها می‌توانیم کلاس پیش بینی شده را به دست آوریم.

۴ نتایج

برای این پروژه باید قسمتی از دادگان آموزش یا همان تست را نیز در اختیار داشته باشیم. این مجموعه با لیبل زدن ۱۰۰۰ خط از opensubtitle فراهم شد. این کار امکان بررسی دقت مدل ساخته شده را ممکن می‌سازد.

همانطور که پیشتر هم اشاره شد، ما برای این پروژه از دیتاست SwDA استفاده خواهیم کرد. این دیتاست مربوط به مکالمات تلفنی می‌باشد که دارای ویژگی‌های خوبی برای تسک ما خواهد بود. یکی از این ویژگی‌ها در ستون "caller" قابل مشاهده است. این ستون با استفاده از اختصاص دادن یک لیبل A یا B به هر گوینده نمایش داده می‌شود. بدین صورت که در صورت A بودن جمله کنونی، اگر گوینده عوض شود، جمله بعدی با لیبل B خواهد آمد و برعکس. هم‌چنین اگر جمله کنونی و جمله بعدی هر دو مربوط به یک گوینده باشند، هر دو A و یا هر دو B میگیرند. یکی از نکاتی که در این دیتاست مطرح می‌باشد این است که شروع هر مکالمه با A خواهد بود، پس حتما باید بررسی کنیم که در کدام سطرها مکالمه در حال تغییر کردن است و مقدار این "caller" در این فیلد را با آخرین جمله از مکالمه قبلی بررسی نکنیم. پس اگر این ویژگی ذکر شده در این دیتاست را بخواهیم به ۰ و ۱ تبدیل کنیم برچسب‌ها به صورت جدول ۳ درمی‌آید.

A	B	B	A	B	A	A	A
۱	۱	۰	۱	۱	۱	۰	۰

جدول ۳: مثالی از نحوه تبدیل لیبل‌های موجود در SwDA به لیبل‌های قابل استفاده

همانطوکه پیشتر نیز اشاره شد، ما از این دیتاست برای ساختن مدلی استفاده خواهیم کرد که ما را قادر به تگ زدن OpenSubtitle کند. اما برای ارزیابی مدل ساخته شه باید حتما تعدادی از جملات این دیتاست را به صورت دستی تگ بزنیم. این جملات هر چه بیشتر باشند بهتر است. نکته قابل ذکر این است که ما در مدل خود از ۱۹۰ هزار داده برای آموزش و حدود ۱۲ هزار داده برای validation استفاده کرده ایم. هم‌چنین اینکه تست نهایی به دلیل عدم ارتباط ۱۰۰۰ خط لیبل زده شده OpenSubtitle فقط بر روی ۳۳۰ نمونه که به ترتیب قرار گرفته بودند صورت گرفت. این لیبل ها به صورت "new" و "continue" در دیتاست قابل مشاهده هستند.

جدول‌های ۴و۵و۶ شامل نتیجه نهایی بر روی مدل‌های ذکر شده در بخش قبل به همراه دیتاست‌های مورد بررسی می‌باشد.

۵ تحلیل

در جداول ۴ و ۵ نتایج استخراج شده به کمک Mallet[۶] دیده می‌شود. این دقت‌ها توسط رده‌بندهای خطی به دست آمده اند. همان‌طور که از جدول شماره ۶ قابل مشاهده است، مدل‌های LSTM بهتر از بقیه عمل کرده‌اند. البته در این جدول میانگین اعداد در هر قسمت گزارش شده است. نکته بسیار مهم در این‌جا این است که به دلیل کم بودن تعداد داده‌های تست بر روی مجموعه دادگان OpenSubtitle دقت‌ها به ازای هر بار اجرا به شدت تغییر می‌کنند و اصلا stable نمی‌باشند. پس ممکن است نتایج گرفته شده از این قسمت با افزایش این داده‌ها تغییر کند و رفتار جدیدی از خود نشان دهند. در صورت مشاهده نمودار تغییر دقت و ارور بر حسب ایپاک در فایل ژوپیتر، می‌بینیم مدلی که فقط بر پایه لایه‌های Dense بوده است، به شدت و ناچارا overfit می‌شود ولی در تست بر روی دادگان OpenSubtitle به خوبی عمل می‌کند. دلیل این امر می‌تواند سادگی زیاد این مدل باشد. دیده می‌شود که مدل‌های با لایه Attention از دقت بیشتری بر روی OpenSubtitle برخوردار هستند. این دقت می‌تواند برگرفته از کلماتی باشد که در داده‌های تست از اهمیت بیشتری برخوردار بوده‌اند و لایه Attention به خوبی آنها را تشخیص داده است.

[۶] http://mallet.cs.umass.edu/

کد ویژگی	دقت آموزش	دقت آزمون
A	۰/۷۴	۰/۶۶
B	۰/۷۰	۰/۶۶
C	۰/۷۰	۰/۶۷
D	۰/۸۱	۰/۶۷
E	۰/۴۵	۰/۴۵
F	۰/۴۵	۰/۴۵
G	۰/۴۶	۰/۴۶
H	۰/۴۷	۰/۴۶
I	۰/۷۳	۰/۶۸
J	۰/۷۴	۰/۷۱
K	۰/۷۵	۰/۷۰
L	۰/۷۷	۰/۷۶
M	۰/۷۸	۰/۷۵

جدول ۴: آموزش و آزمون داده‌ها بر روی دیتاست SwDA توسط رده‌بند Bayes Naïve

کد ویژگی	دقت آزمون روی SwDA	دقت آزمون روی OpenSubtitle
A	۰/۷۵	۰/۵۱
B	۰/۷۴	۰/۶۳
C	۰/۷۰	۰/۵۵
D	۰/۷۴	۰/۵۴
E	۰/۶۸	۰/۶۰
F	۰/۷۱	۰/۶۲
G	۰/۶۷	۰/۶۰
H	۰/۷۱	۰/۶۳
I	۰/۷۵	۰/۵۴
J	۰/۷۳	۰/۶۷
K	۰/۷۶	۰/۶۲
L	۰/۷۷	۰/۶۵
M	۰/۸۱	۰/۶۰
N	۰/۸۲	۰/۶۲

جدول ۵: آموزش مدل بر روی SwDA و آزمون داده‌های هر دو دیتاست توسط Sequence Tagging

	SwDA				OpenSubtitle	
Method	**Train Acc**	**Train Loss**	**Test Acc**	**Test F1**	**Test Acc**	**Test F1**
BERT	۷۵/۴۶	۰/۵۸۶	۷۲/۶۰	۷۲/۶۵	NaN	NaN
Dense only	۸۱/۹۹	۰/۳۸۱	۷۸/۳۶	۸۰/۷۱	۵۷/۵۹	۷۰/۲۲
LSTM	۸۱/۵۲	۰/۴۰۳	۸۲/۳۲	۸۳/۷۶	۴۹/۰۵	۵۹/۲۴
LSTM with Attention	۸۰/۵۶	۰/۴۲۲	۸۱/۳۰	۸۲/۶۶	۵۱/۲۷	۶۰/۳۱
BiLSTM	۸۱/۴۵	۰/۴۰۴	۸۱/۹۳	۸۳/۱۵	۵۰/۶۳	۶۱/۵۸
BiLSTM with Attention	۸۱/۶۲	۰/۴۰۲	۸۲/۲۱	۸۳/۰۴	۵۶/۶۴	۶۶/۳۴

جدول ۶: نتایج حاصل از اجرای مدل‌های مختلف بر روی هر دو دیتاست

منابع

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[2] John Godfrey and Edward Holliman. Switchboard-1 release 2 ldc97s62. *DVD. Philadelphia: Linguistic Data Consortium*, 1993.

[3] Yuichi Ishimoto, Takehiro Teraoka, and Mika Enomoto. End-of-utterance prediction by prosodic features and phrase-dependency structure in spontaneous japanese speech. In *Interspeech*, pages 1681–1685, 2017.

[4] Chaoran Liu, Carlos Toshinori Ishi, and Hiroshi Ishiguro. Turn-taking estimation model based on joint embedding of lexical and prosodic contents. In *Interspeech*, pages 1686–1690, 2017.

[5] Angelika Maier, Julian Hough, and David Schlangen. Towards deep end-of-turn prediction for situated spoken dialogue systems. *Proceedings of INTERSPEECH 2017*, 2017.

[6] Ryo Masumura, Taichi Asami, Hirokazu Masataki, Ryo Ishii, and Ryuichiro Higashinaka. Online end-of-turn detection from speech based on stacked time-asynchronous sequential networks. In *Interspeech*, volume 2017, pages 1661–1665, 2017.

[7] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. *http://mallet. cs. umass. edu*, 2002.

[8] Colin Raffel and Daniel P. W. Ellis. Feed-forward networks with attention can solve some long-term memory problems, 2015.

[9] Idris Yusupov and Yurii Kuratov. Nips conversational intelligence challenge 2017 winner system: Skill-based conversational agent with supervised dialog manager. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3681–3692, 2018.