به نام خدا

پروژه درس پردازش زبان طبیعی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف

زمستان 1400

موضوع: پیشنهاد شعر براساس متن

گروه HMA

امیرحسین عاملی

محمدرضا یزدانی‌فر

حسین خلیلی

محمد مظفری

چکیده

زبان فارسی دارای تاریخچه زیاد و شاعران فارسی زبان جز بزرگترین شاعران تاریخ هستند. به عنوان یک فارسی زبان وظیفه ماست تا در حفظ و گسترش زبان و فرهنگ خود بکوشیم. انجام این پروژه به عنوان یک پردازش بر روی شعرهای فارسی میتواند قدمی کوچک در این راستا باشد.

هدف این پروژه تولید یک وب اپلیکیشن است که در آن کاربر متن دلخواه خود را وارد نموده و اپلیکیشن ما به شکل هوشمند شعری متناسب با متن وارد شده کاربر به او پیشنهاد دهدو کاربر از زیبایی‌های زبان فارسی و اشعار غنی فارسی لذت ببرد.

همچنین این پروژه می‌تواند استفاده‌های تحقیقاتی نیز داشته و کمک کند متناسب با مضمون مورد نظر شعر و یا حتی بیت و مصراع دسته‌بندی شود و یک آرشیو محتوایی غنی از اشعار فارسی ایجاد شود.

همچینین از دیگر دست آوردهای این پروژه می‌توان به تولید داده‌ی ارزیابی برای ارزیابی روش‌های پیشنهادی نام برد که می‌تواند در آینده برای مقایسه دیگر مدل‌ها نیز استفاده شود و کارهای آتی با این پروژه مقایسه شوند.

این وب اپلیکیشن از این لینک در دسترس است.

مقدمه

شعر فارسی تاریخ کهنی دارد از حدود ۱۱۰۰ سال قبل شاعران پارسی زبان به سردودن شعر به شکل امروزی پرداخته‌اند. شاعران بزرگی همچون حافظ، سعدی، فردوسی، مولودی، خیام و.. کمک شایانی به غنی‌تر شدن ادبیات فارسی کرده و بر زیبایی‌های ادبیات پارسی افزوده اند.همچنین اشعار پارسی سرچمشه گرفته از فرهنگ پارسی خود به گسترش این فرهنگ غنی میسر شده اند. از این رو حفظ و نگهداری از این آثار امر بسیار مهمی است که نسل به نسل بر دوش ما قرار گرفته است.

با توجه به پیشرفت روز افزن هوش مصنوعی و قدرت‌های پردازشی که هر روزه بیشتر و بیشتر می‌شوند،استفاده از این ابزار قدرتمند در زمینه ادبیات پارسی ، مشهود می‌شود. وقتی که ابزارهای هوش مصنوعی توانسته‌اند تصاویر خلق کنند که هرگز وجود نداشته و با تصاویر واقعی اندکی تفاوت ندارند و یا توانسته اند زبان انسان را به خوبی درک کنند و به راحتی با انسان صحبت کنند چرا از این ابزار فوق‌العاده در راستای حفظ ادبیات فارسی استفاده نکنیم؟.

استفاده از هوش مصنوعی در پردازش اشعار پارسی جدا از این که می‌تواند به حفظ و نشر اشعار غنی پارسی کمک کند؛ می‌تواند باعث پیشرفت هوش‌مصنوعی در تحلیل زبان پارسی شود و به استفاده هوش مصنوعی در زبان پارسی برای نسل‌های بعد کمک فراوانی کند و عقب ماندگی این بخش از هوش مصنوعی نسبت به دیگر زبان‌های زنده دنیا کم و کمتر شود.

از این رو ایجاد یک اپلیکیشن بر بستر هوش مصنوعی که بتواند متناسب با متن ارسالی، شعر و یا بیت و یا مصراعی از اشعار پارسی پیدا نموده و به کاربر نشان دهد ارزشمند می‌باشد زیرا در وهله اول باعث علاقه‌مند به شعر فارسی شده و همچنین می‌تواند با کمک این ابزار متناسب با هر مضمون شعری را پیدا کرد و دسته بندی محتوایی بر روی آرشیو اشعار فارسی انجام داد. و همچنین در وهله دوم کمک شایانی به پیشرفت هوش مصنوعی در کاربرد زبان فارسی می‌کند تا لحظه به لحظه از قافله‌ی دیگر زبان‌های زنده‌ی دنیا جا نمانیم.

شایان ذکر است که این کار چالش‌های فراوانی دارد که در ادامه به روش‌های پیشین و چالش‌های این کار می‌پردازیم

کارهای مشابه

از جمله ساده ترین روش‌های بازیابی اطلاعات می‌توان بازیابی به کمک روش TF-IDF اشاره کرد. در این روش که تعداد تکرار کلمات در هر داکیومنت شمرده شده و متناسب با فرکانس تکرار کلمه در هر داکیومنت و تعداد داکیومنت‌هایی که این کلمه را دارا می‌باشد یک ماتریش تشکیل داده که تعداد سطرهای آن برابر با تعداد داکیومنت‌ها و تعداد ستون‌های آن متناسب با تعداد کلمات دیکشنری ما هستش.

استفاده از این روش برای بازیابی اطلاعات بدین شکل است که در ابتدا تمامی داکیومنت‌هایی که قرار است در آن‌ها جست و جو ضورت بگیرد توسط TF-IDF مدل می‌شوند و هنگامی که یک query داده می‌شود با کمک مدل TF-IDF بردار آن محاسبه شده و فاصله کسینوسی این بردار با تمام سطرهای این ماتریس TF-IDf محاسبه می‌شود و آن سطری که کمترین فاصله را دارا می‌باشد داکیومنت مورد نظر ما خواهد بود.

استفاده از این روش در مسئله ما دارای نقص‌هایی می‌باشد. از جمله اینکه تنها به کلمات و کلید واژه‌ها توجه میکند و اگر در متن کوئری ما کلمه "عشق" زیاد تکرار شده باشد به دنبال متونی می‌رود که کلمه عشق در آن ها بسیار تکرار شده باشد ولی ممکن است متن ما این باشد : "عشق وجود ندارد و عشق افسانه‌ای بیشتر نیست" ولی شعر مرتبط هیچ ربطی به مضمون متن ما نداشته است.

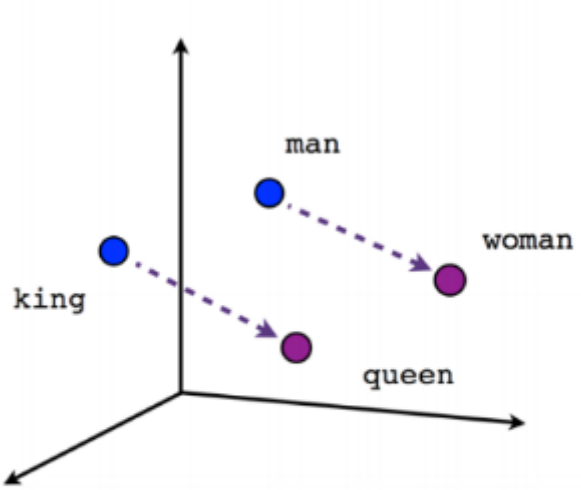
برای حل مشکل و همچنین جست و جوی معنایی دقیق‌تر باید به سراغ مدل‌هایی رفت که مضمون متن ما را به خوبی مدل کرده و متناسب با آن شعر مورد نظر ما را پیشنهاد دهد.

مدل TF-IDF به عنوان یکی از مدل‌ها در اپلیکیشن ما آمده است صرفا به منظور اینکه روش بیس‌لاین نیز پیاده شده و خروجی آن مشاهده شود.

معرفی روش‌ها

روش Doc2Vec:

ما در درس با روش بازنمایی Word2Vec آشنا شدیم. این روش برای اولین بار در سال 2013 توسط آقای Mikolov و همکارانش ارائه شد. در این الگوریتم، هدف بدست آوردن یک بازنمایی برای کلمات به نحوی بود که کلمات با معنی و مرتبط در کنار یکدیگر باشند و همچنین روی این بردارهای بازنمایی بتوانیم عملیات جبری نیز انجام دهیم. برای مثال با جمع بردار بردار کلمات King و Woman و تفریق بردار کلمه Man به کلمه Queen برسیم.



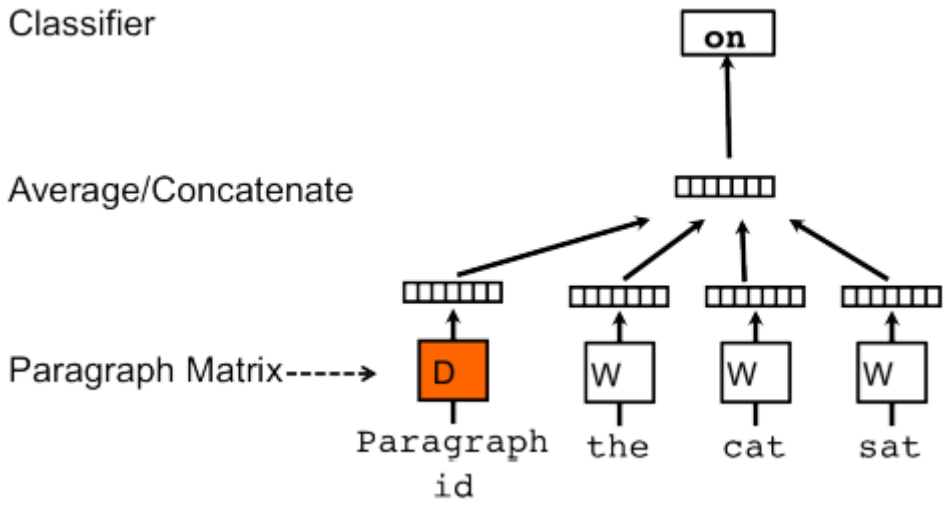
Male-Female

نتایج بدست آمده از این روش مطابق موردی بود که بالا مطرح شد. مسئله ای که با این روش قابل حل نبود، بدست آوردن بازنمایی برای جملات بود که با استفاده از این روش تنها می توانستند میانگین بردار بازنمایی کلمات را بدست آورند. به همین دلیل در سال 2014 مجددا آقای Mikolov برپایه روش Word2vec روش دیگری به نام Doc2vec را

ارائه نمود. همانطوری که از اسم روش مشخص است، در این روش یک بردار بازنمایی این بار برای هر document بدست می آمد. در این مقاله دو روش با استفاده از روش های مختلف Word2vec ارائه شد.

روش PV-DM

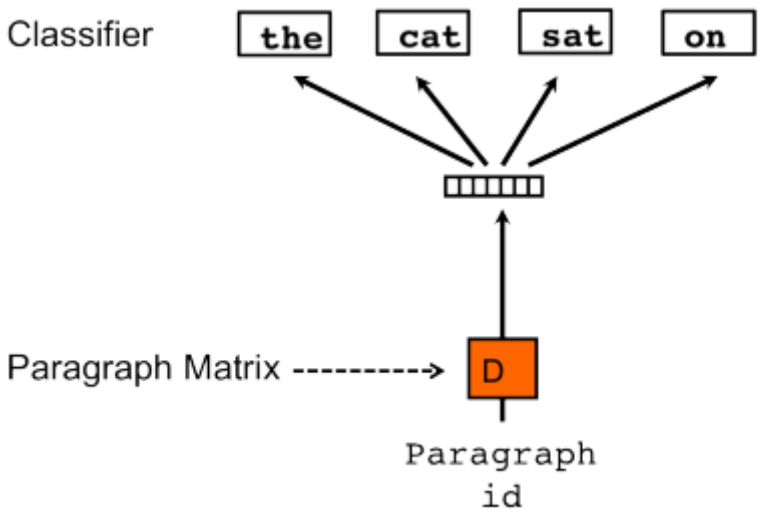
در این روش یک بردار بازنمایی دیگری با دادن ورودی ID سند مورد نظر به CBOW آموزش داده می شود:



به این ترتیب به ازای هر document در فرایند آموزش یک بردار بازنمایی برای هر سند نیز یاد گرفته می شود که میتواند در downstream tasks استفاده شود. این روش که نام آن Distributed Memory version of Paragraph Vector می باشد، از این رو نامگذاری شده است که این بازنمایی به تعبیری دربردارنده اطلاعاتی است که در کلمات مدل نمی باشد ولی مرتبط با Topic و موضوع document برای پیش بینی کلمه وسط می‌باشد.

روش PV-DBOW

روش دوم معرفی شده برپایه روش Skip-gram می باشد:



در این روش تنها با دادن بردار بازنمایی ورودی مدل باید کلمات مجاور هم در document را پیش بینی نماید. این روش برخلاف روش قبلی هم آموزش سریعی دارد و هم حافظه کمتری را مصرف می‌کند. چون نیاز ندارد که برای هر کلمه نیز یک بردار بازنمایی یاد بگیرد. نام این روش در مقاله Distributed Bag of Words version of Paragraph Vector می باشد. روش های دیگری نیز برای doc2vec مانند اضافه کردن یک بازنمایی دیگر با استفاده از tag ها نیز معرفی شده است ولی به دلیل ناسازگار بودن با دادگان و مسئله ما، در این پژوهش بررسی نشده است.

برای این روش هردو مدل، یک بار با اندازه بردار بازنمایی 100 و یک بار دیگر با اندازه بردار 300 در مجموع 4 مدل آموزش و مورد ارزیابی قرار گرفته شده است که در بخش نتایج می توانیم عملکردهای آنها را مشاهده نماییم.

عملکرد این مدل در inference کمی متفاوت با مدل های دیگر می باشد. در این مدل برای یک query و document جدید یک fine-tuning روی مدل یادگرفته شده به تعداد epoch کم می شود و این چنین inference انجام می شود.

مدل های مبتنی بر BERT

مدل 1: استفاده از BERT ساده

در این حالت از BERT به عنوان یک Sentence Transformer استفاده میشود به این صورت که یک لایه ترنسفورمر برت بعلاوه یک لایه Pooling خواهیم داشت و مجموع این دو Sentence Transformer ما را میسازند. در این مدل از BERT از قبل آموزش داده شده زیر استفاده میکنیم که در Hugging Face موجود است.

به این ترتیب یک شبکه خواهیم داشت که با دریافت یک بیت شعر به عنوان ورودی در خروجی یک بازنمایی از آنرا به ما میدهد. بنابراین در زمان تست کوئری مورد نظر را نیز با استفاده از همین شبکه بازنمایی میکنیم. سپس cosine similarity این بازنمایی با بازنمایی تمام بیتها را محاسبه میکنیم. K تا از بیتهایی که بیشترین شباهت را داشته باشند را به عنوان خروجی بازمیگردانیم.

مدل 2: استفاده از BERT به علاوه Cross Encoder

در این روش برای اینکه بتوانیم با دقت بهتری بازیابی را انجام دهیم به این صورت عمل میکنیم.
فرض کنید میخواهیم K بیت به عنوان خروجی برگردانیم. ابتدا با استفاده از روش توضیح داده شده در قسمت قبل M بیت با شباهت بیشتر را پیدا میکنیم به گونه ای که $M>K$. سپس با استفاده از یک مدل Cross Encoder شباهت هر بیت از این M بیت را با کوئری می یابیم. Cross Encoder به این صورت عمل میکند که دو جمله به عنوان ورودی دریافت میکند و در خروجی میزان شباهت را میدهد. در پروژه از Cross Encoder آماده زیر استفاده میکنیم.

روش محاسبه میزان شباهت پیچیده تر از محاسبه یک شباهت کسینوسی است. حال از بین این M بیت K بیتی که شباهت آنها (شباهت محاسبه شده توسط Cross Encoder) با کوئری بیشترین است را برمیگردانیم.

مدل 3: استفاده از BERT فاینتون شده

این مدل دقیقا شبیه مدل 1 است با این تفاوت که از BERT داده شده مستقیما استفاده نمیشود. بلکه ابتدا این مدل لود میشود و سپس بر روی تمام دادگان شعری fine-tune میشود. به این ترتیب مدل بدست آمده برای دادههای شعری مناسب تر خواهد بود. ادامه کار دقیقا مشابه مدل 1 است.

مدلهای مبتنی بر درستنمایی جستجو

مدل unigram به ازای هر سند

در این روش به ازای هر سند، یک مدل unigram در نظر میگیریم. تخمین بیشینه درستنمایی پارامتر احتمال هر کلمه، به ازای هر سند، برابر با تکرار آن کلمه تقسیمبر تعداد کل کلمات آن سند است. این موضوع در معادلهی زیر واضحتر میشود. در رابطهی زیر میگوید احتمال یک کلمهی خاص q در سند d برابر با تعداد تکرارهای آن کلمه در آن سند تقسیمبر تعداد کل کلمات آن سند است. این همان تخمین بیشینه درستنمایی مدل unigram برای آن سند است.

$$p_d(q) = \frac{N_{d,q}}{N_d}$$

برای بازیابی اطلاعات، درستنمایی یک عبارت را به شکل ضرب درستنمایی تک تک کلمات آن مینویسیم:

$$p(Q \mid D) = \prod_{q \in Q} p(q \mid D)$$

به این وسیله، درستنمایی عبارت برای تمام اسناد محاسبه میشود و اسناد برحسب درستنمایی مرتب میشوند.

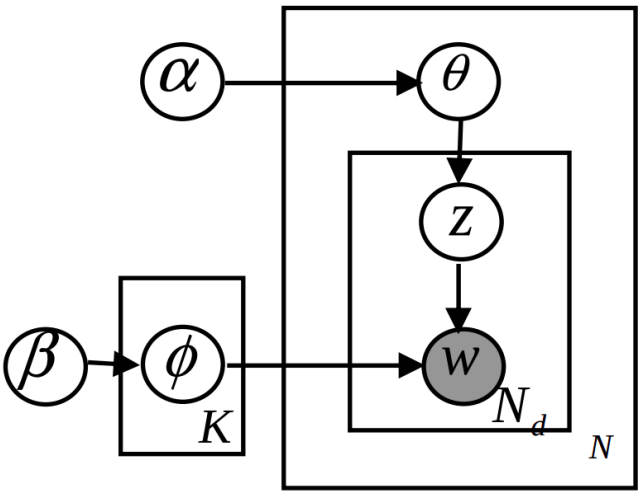
همچنین برای جلوگیری از مشکلات محاسبات عددی از تکنیک نرم کردن لاپلاس استفاده میکنیم:

$$p_d(q) = \frac{N_{d,q} + 0.001}{N_d + 0.001V}$$

ما فرض کردیم که هر کلمه حداقل یک هزارم بار در یک سند ظاهر می‌شود.

مدل LDA

از آنجا که LDA یک مدل احتمالاتی است که به ازای هر سند یک توزیع احتمال لغات تعریف می‌کند، می‌توان برای هر کلمه در یک سند درستنمایی محاسبه کرد. لذا مشابه روش قبل عمل می‌کنیم. اگر مدل احتمالاتی تخصیص نهان دریشله را بصورت زیر در نظر بگیریم:



درستنمایی بشکل زیر محاسبه می‌شود:

$$P_{lda}(w \mid d, \hat{\theta}, \hat{\phi}) = \sum_{z=1}^{K} P(w \mid z, \hat{\phi}) P(z \mid \hat{\theta}, d)$$

که در آن $\hat{\theta}$ و $\hat{\phi}$ توزیع‌های پسین متغییرهای تصادفی متناظرشان هستند.

داده

برای آموزش مدل های مد نظر برای این مسئله از مجموعه داده شعرای مختلف استفاده می‌شود. در زیر آدرس یک مخزن حاوی اشعار شعرای فارسی آورده شده که میتوانند برای پروژه مورد استفاده قرار گیرند.

https://github.com/amnghd/Persian_poems_corpus

در این مخزن اشعار مربوط به 48 شاعر مختلف آورده شده است.

لیست شاعران مخزن :

ابوسعید ابوالخیر امیرمعزی، اوحد الدین انوری، فخرادلین اسعد گرگانی ، اسدی توسی ، عطار نیشابوری، افضل الدین کاشانی ، شیخ بهایی، ملک اشعرای بهار، بیدل دهلوی، فخرالدین عراقی، فرخی یزدی، فردوسی فیض کاشانی، قاآنی شیرازی، عبدالقادر گیلانی، حافظ شیرازی، هاتف اصفهانی، بدرالدین هلالی، محد اقبال لاهوری ، جامی، کمال الدین اسماعیل، خاقانی ، خاجوی کرمانی، عمر خیام ، امیر خسرو ، منوچهری دامغانی ، مولانا، ناصرخسرو ، نزاری قهستانی، عبید زاکانی ، عنصری بلخی، عرفی شیرازی، اوحدی مراغه ای ، پروین اعتصامی، رهی معیری، رضی ادین آرتیمانی ، رودکی ، سعدی شیرازی، صائب تبریزی، مسعود سعد سلمان ، سنایی غزنوی، صیف فرغانی، محمود شبستری ، شاه نعمت الله ولی ، شهریار، وحشی بافقی ، ظهیرالدین فاریابی.

نتایج

تولید داده‌های ارزیابی

این بخش که خود یکی از دست‌آوردهای این پروژه می‌باشد بدین شکل به دست آمده است:

مراحل تولید داده‌ی ارزیابی به شرح زیر است:

۱) نمونه گرفتن از n-gramهای لمتایز شده اشعار براساس idf

۲) جایگزینی کلمات با استفاده از پیشنهادهای فضای امبدینگ

۳) چک کردن اینکه کوئری جدید جواب surface form زیادی نداشته باشد

۴) دو نفر دیگر مستقلا به درست بودن کوئری‌هایی که تولید شده امتیاز رد و قبول می‌دهند و میزان توافق آنها سنجیده می‌شود. در موارد اختلافی، ۵۰ درصد موارد بصورت تصادفی انتخاب می‌شود.

روش درست‌نمایی مدل unigram به ازای هر سند

K	Precision @ K
1	0.54
21	0.55
41	0.59
61	0.63
81	0.65
101	0.68

روش درست‌نمایی مدل LDA

K	Precision @ K
101	0.36

روش FastText با استفاده از Pre-Training

K	Precision @ K
1	0.06
21	0.08
41	0.11
61	0.14
81	0.17
101	0.19

روش FastText بدون استفاده از Pre-Training

K	Precision @ K
1	0.03
21	0.06

41	0.09
61	0.10
81	0.11
101	0.13

روش مبتنی بر BERT اول (برت ساده)

K	Precision @ K
1	0.140
21	0.081
41	0.065
61	0.054
81	0.046
101	0.044

روش مبتنی بر BERT دوم (برت با استفاده از CrossEncoder)

K	Precision @ K
1	0
21	0
41	0
61	0

81	0
101	9.9e-5

روش مبتنی بر BERT سوم(برت Finetune شده)

K	Precision @ K
1	0.270
21	0.129
41	0.100
61	0.082
81	0.072
101	0.064

روش PV-DBOW با اندازه بردار 300

K	Precision @ K
20	0.21364
40	0.27462
60	0.30354
80	0.3165
100	0.33488

همچنین مقدار MRR میانگین برابر 0.05269 می باشد.

روش PV-DBOW با اندازه بردار 100

K	Precision @ K
20	0.13029
40	0.16003
60	0.1787
80	0.19168
100	0.21033

همچنین مقدار MRR میانگین برابر 0.02659 می باشد.

روش PV-DM با اندازه بردار 300

K	Precision @ K
20	0.04359
40	0.05194
60	0.05836
80	0.06448
100	0.06722

همچنین مقدار MRR میانگین برابر 0.018768 می باشد.

روش PV-DM با اندازه بردار 100

K	Precision @ K
20	0.000505

40	0.000252
60	0.000246
80	0.000246
100	0.000246

همچنین مقدار MRR میانگین برابر 0.00026 می باشد.

نتیجه گیری

همانطور که از نتایج برمی‌آید، مدل درست‌نمایی کوئری بهترین نتیجه را دارد. دلیل این موضوع آنست که علی‌رغم سادگی این مدل و لحاظ نکردن مفهوم عبارات، این مدل به عین کلمات بسیار حساس است و به دلیل ناپارامتری بودن عملکرد خوبی دارد و می‌تواند اسناد را به خوبی از هم جدا کند. مدل تخصیص نهان دریشله، به نسبت خود عملکرد خوبی دارد. اما این مدل به دادگان با تعداد زیاد حساس است. مدل FastText به دلیل تعداد پارامترهای زیاد و پیچیدگی‌های همجواری لغات در شعر توفیق چندانی نداشت. همچنین مشاهده می‌شود که استفاده از مدل از پیش آموزش دیده، تاثیر قابل مناسبی در فرآیند یادگیری دارد.

همانطور که در بخش نتایج دیدیم از بین مدل های مبتنی بر برت مدلی که بر روی دیتا fine-tune شده بود به بهترین نتیجه دست یافت. این موضوع به دلیل این است که مدل fine-tune شده اطلاعات بهتری نسبت به حوزه مسئله دارد نسبت به مدل برت ساده. مدل برت ساده بر روی متون فارسی آموزش دیده و fine-tune کردن آن بر روی دیتای شعری نتیجه را بهتر میکند. مدل مبتنی بر Cross Encoder ضعیف ترین نتیجه را داشته که بسیار ناامید کننده است. در صورتی که انتظار می رفت این مدل نتیجه خوبی داشته باشد. دلیل این امر نیز احتمالا به خاطر Cross Encoder

مورد استفاده است که ممکن است به خوبی آموزش ندیده باشد. در صورت استفاده از Cross Encoder بهتر احتمالا نتیجه نیز بهتر خواهد شد.

مدل های مبتنی بر Doc2vec نیز نتایج به نسبت خوبی بدست آوردند که بهترین آن مدل PV-DBOW با اندازه بردار بازنمایی 300 بود. طبق نتایج هرچه اندازه بردار بیشتر باشد نتیجه بهتر می باشد. دلیل این امر این است که در حالتی که اندازه بردار کم می باشد پدیده Underfitting رخ می دهد. همچنین دلیل اینکه PV-DBOW خیلی بهتر از PV-DM می شود این است که در این مدل بردار بدست آمده برای document دارای اطلاعات زیادی در مورد کل document می باشد که توانسته است در فرایند آموزش عملکرد خوبی در پیش بینی کلمات مجاور document داشته باشد.