Cross-genre Document Retrieval: Matching between Conversational and Formal Writings

Tomasz Jurczyk
Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
tomasz.jurczyk@emory.edu

Jinho D. Choi
Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
jinho.choi@emory.edu

Abstract

This paper challenges a cross-genre document retrieval task, where the queries are in formal writing and the target documents are in conversational writing. In this task, a query, is a sentence extracted from either a summary or a plot of an episode in a TV show, and the target document consists of transcripts from the corresponding episode. To establish a strong baseline, we employ the current state-of-the-art search engine to perform document retrieval on the dataset collected for this work. We then introduce a structure reranking approach to improve the initial ranking by utilizing syntactic and semantic structures generated by NLP tools. Our evaluation shows an improvement of more than 4% when the structure reranking is applied, which is very promising.

1 Introduction

Document retrieval has been a central task in natural language processing and information retrieval. The goal is to match a query against a set of documents. Over the last decade, advanced techniques have emerged and provided powerful systems that can accurately retrieve relevant documents (Blair and Maron, 1985; Callan, 1994; Cao et al., 2006). While the retrieval part is crucial, proper ranking of the retrieved documents can significantly improve the overall user satisfation by putting more relevant documents at the top (Baliński and Daniłowicz, 2005; Yang et al., 2006; Zhou and Wade, 2009). Many previous works provide strong baselines for unstructured text retrieval and ranking problems; however, these systems usually assume a homogeneous domain for queries and target documents.

Due to the spike of applications that are required to maintain the conversation, dialog data has recently become a popular target among researchers. The work in this field concerns problems such as learning facts through conversation (Fernández et al., 2011; Williams et al., 2015; Hixon et al., 2015) or dialog summarization (Oya and Carenini, 2014; Misra et al., 2015). More recent work in this field has focused on several inter-dialogue tasks (Xu and Reitter, 2016; Kim et al., 2016; He et al., 2016). To the best of our knowledge our work is the first, where the cross-genre document retrieval is analyzed based on conversational and formal writings.

This paper analyzes the performance of state-of-the-art retrieval techniques targeting TV show transcripts and their descriptions. We first collect a dataset comprising transcripts from a popular TV show and their summaries and plots (Section 3). We then establish a solid baseline by adapting an advanced search engine and implement structure reranking to improve the initial ranking from the search engine (Section 4). Our evaluation shows a 4% improvement, which is significant (Section 5).

2 Related work

Information extraction for dialogue data has already been widely explored. Yoshino et al. (2011) presented a spoken dialogue system that extracts predicate-argument structures and uses them to extract facts from news documents. Flycht-Eriksson and Jönsson (2003) developed a dialogue interaction process of accessing textual data from a bird encyclopedia. An unsupervised technique for meeting summarization using decision-related utterances has been presented by Wang and Cardie (2012). Gorinski and Lapata (2015) studied movie script summarization. All the aforementioned work uses the syntactic and semantic relation extraction and thus is similar to ours; however, it is distinguished in a way that it lacks a cross-genre aspect.

Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems, pages 48–53
Copenhagen, Denmark, September 8, 2017.

Dialogue		Summary + Plot
Joey	One woman? That's like saying there's only one flavor of ice cream for you. Lemme tell you something, Ross. There's lots of flavors out there.	Joey compares women to ice cream. (*S*)
Ross Rachel	You know you probably didn't know this, but back in high school, I had a, um, major crush on you. I knew.	Ross reveals his high school crush on Rachel. (*S*)
Chandler Joey	Alright, one of you give me your underpants. Can't help you, I'm not wearing any.	Chandler asks Joey for his underwear, but Joey can't help him out as he's not wearing any. (*P*)

Table 1: Three manually curated examples of dialogues and their descriptions.

3 Data

The Character Mining project provides transcripts of the TV show, *Friends*; transcripts from 8 seasons of the show are publicly available in the JSON format,[1] where the first 2 seasons are annotated for the character identification task (Chen and Choi, 2016). Each season consists of episodes, each episode contains scenes, each scene includes utterances, where each utterance comes with the speaker information.

For each episode, the episode summary and plot are first collected from fan sites,[2] then sentence segmented by NLP4J,[3] the same tool used for the provided transcripts. Generally, summaries give broad descriptions of the episodes, whereas plots describe facts within individual scenes. Finally, we create a dataset by treating each sentence as a query and its relevant episode as the target document. Table 2 shows the distributions of this dataset.

Dialogue		Summary + Plot	
# of episodes	194	# of queries	5,075
# of tokens	897,446	# of tokens	119,624

Table 2: Dialogue, summary, and plot data.

4 Structure Reranking

For each query (summary or plot) in the dataset, the task is to retrieve the document (episode) most relevant to the query. The challenge comes from the cross-genre aspect: how to retrieve documents in dialogues given the queries in formal writing. This section describes our structure reranking approach that significantly outperforms an advanced search engine, Elasticsearch[4].

4.1 Relation Extraction

Since our queries and documents appear very different on the surface level (Table 1), relations are first extracted from them and matching is performed on the relation level, which abstracts certain pragmatic differences between these two types of writings. All data are lemmatized, tagged with parts-of-speech and named entities, parsed into dependency trees, and labeled with semantic roles using NLP4J.

A sentence may consist of multiple predicates, and each predicate comes with a set of arguments. A predicate together with its arguments is considered a relation. For each argument, heuristics are applied to extract meaningful contextual words by traversing the subtree of the argument. Our heuristics are designed for the type of dependency trees generated by NLP4J, but similar rules can be generalized to other types of dependency trees. Relations from dialogues are attached with the speaker names to compensate the lack of entity information.

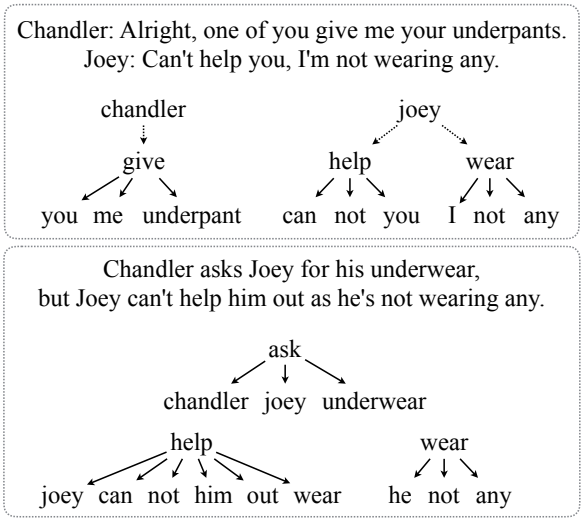

Figure 1: Two sets of relations, from dialogue and plot, extracted from the examples in Table 1.

By extracting relations that comprise only meaningful words, it prunes out much noise (e.g., disfluency), which allows the system to retrieve relevant documents with higher precision. While our relation extraction is based on the sentence level, it can be extended to the document level by adding coreference relations, which we will explore in the future.

[1] nlp.mathcs.emory.edu/character-mining
[2] friends-tv.org, friends.wikia.com
[3] github.com/emorynlp/nlp4j
[4] https://www.elastic.co/

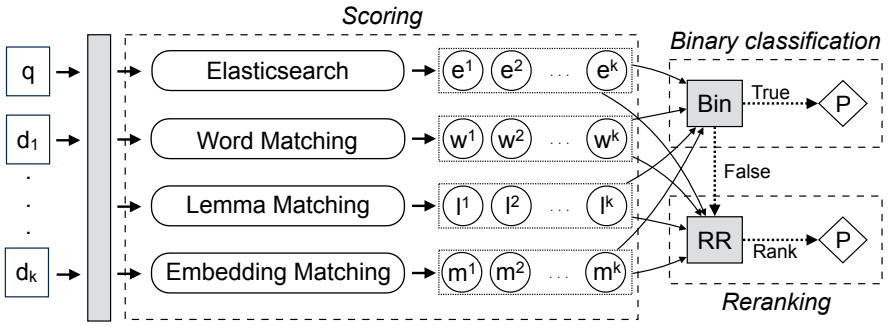

Figure 2: The overview of our structure reranking. Given documents $d_1, \ldots, d_k$ and a query q, 4 sets of scores are generated: the Elasticsearch scores and the matching scores using 3 comparators: *word*, *lemma*, and *embedding*. The binary classifier `Bin` predicts whether the highest ranked document from Elasticsearch is the correct answer. If not, the system `RR` reranks the documents using all scores and returns a new top-ranked prediction.

4.2 Structure Matching

All relations extracted from dialogues are stored in an inverted index manner, where words in each relation are associated with the relation and the episode in which the relation occurs. Algorithm 1 shows how our structure matching works. Given a list of documents retrieved from the index based on a query q, it first initializes scores for all documents to 0. For each document d_i, it compares each relation r^q from q to relations extracted from d_i. The relation r from d_i is kept within R^d if it has at least one word that overlaps with r^q. For each relation $r^d \in R^d$, the comparator function returns the matching score between r^d and r^q. The maximum matching score is added to the overall score of this document. This procedure is repeated; finally, the algorithm returns the overall matching scores for all documents.

Input: D: a list of documents, q: a query.
f_r: a function returning all relations.
f_c: a comparator function.
Output: S: a list of matching scores for D.
$S \leftarrow [0 \text{ for } i \in [1, |S|]$
foreach $d_i \in D$ **do**
  foreach $r^q \in f_r(q)$ **do**
    $R^d \leftarrow [r \text{ for } r \in f_r(d_i) \text{ if } |r \cap r^q| \geq 1]$
    $s_m \leftarrow 0$
    foreach r^d *in* R^d **do**
      $s \leftarrow f_c(r^d, r^q)$
      $s_m \leftarrow \max(s_m, s)$
    end
    $S_i \leftarrow S_i + s_m$
  end
end

Algorithm 1: The structure matching algorithm.

The comparator function f_c takes two relation sets, r^d and r^q, and returns the matching score between those two sets. For $word$ and $lemma$, the count of overlapping words between them is used to produce two scores, r^d_s, and r^q_s, normalized by the length of the utterance and the query, respectively. The harmonic mean of the two scores is then returned as the final score. For $embedding$, f_c uses embeddings to generate sum vectors from both sets and returns the cosine similarity of these two vectors.

4.3 Document Reranking

The Elasticsearch scores and the 3 sets of matching scores for the top-k documents (ranked by Elasticsearch) are fed into a binary classifier to determine whether or not to accept the highest ranked document. A Feed Forward Neural Network with one hidden layer of size 15 is used for this classification. If the binary classifier disqualifies the top-ranked document, the top-k documents are reranked by the weighted sums of these scores. A grid search is performed on the development set to find the optimized set of the weights. At last, the system returns the document with the highest reranked scores: $d_i = \arg\max_i(\lambda_e \cdot e_i + \lambda_w \cdot w_i + \lambda_l \cdot l_i + \lambda_m \cdot m_i)$.

5 Experiments

The data in Section 3 is split into training, development and evaluation sets, where queries from each episode are randomly assigned. Two standard metrics are used for evaluation: precision at k (P@k) and mean reciprocal rank (MRR).

Dataset	Summary	Plot	Total
Training	970	3,013	3,983 (78.48%)
Development	97	403	500 (9.85%)
Evaluation	150	442	592 (11.67%)

Table 3: Data split (# of queries).

Model	Development						Evaluation					
	Summary		Plot		All		Summary		Plot		All	
	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
Elastic_{10}	44.33	53.64	46.40	54.97	46.00	54.71	50.67	60.87	46.61	55.06	47.64	56.53
Struct_w	38.14	48.42	34.00	45.11	34.80	45.75	35.33	48.34	35.52	47.08	35.47	47.40
Struct_l	39.18	49.24	34.74	46.29	**35.60**	**46.86**	44.00	55.55	38.01	49.24	**39.53**	**50.84**
Struct_m	35.05	46.71	33.50	44.72	33.80	45.10	36.00	50.14	35.97	46.95	35.98	47.76
Rerank_1	47.42	55.66	48.39	56.10	48.20	56.02	56.67	63.77	50.23	57.99	51.86	59.46
Rerank_λ	50.52	57.66	51.36	57.76	**51.20**	**57.74**	55.33	63.88	50.90	58.47	**52.03**	**59.84**

Table 4: Evaluation on the development and evaluation sets for summary, plot, and all (summary + plot). Elastic_{10}: Elasticsearch with k = 10, $\text{Struct}_{w,l,m}$: structure matching using words, lemmas, embeddings, $\text{Rerank}_{1,\lambda}$: unweighted and weighted reranking.

5.1 Elasticsearch

Elasticsearch is used to establish a strong baseline.[5] Each episode is indexed as a document using the default setting, Okapi BM25 (Robertson et al., 2009), and the TF-IDF based similarity with improved normalization; the top-k most relevant documents are retrieved for each query. While P@1 is less than 50% (Table 5), P@10 shows greater than 70% coverage implying that it is possible to achieve a higher P@1 by reranking results from $k \geq 10$.

k	Development		Evaluation	
	P@k	MRR	P@k	MRR
1	**46.00**	46.00	**47.64**	47.64
5	65.80	53.80	69.26	69.26
10	72.60	**54.71**	74.66	**56.53**
20	78.80	55.13	79.73	56.91
40	83.80	55.31	84.80	57.08

Table 5: Elasticsearch results on (summary + plot).

5.2 Structure Matching

The Struct_* rows in Table 4 show the results based on structure matching (Section 4.2). The highest P@1 of 39.53% is achieved on the evaluation set using lemmas. Although it is about 8% lower than the one achieved by Elasticsearch, we hypothesize that this approach can correctly retrieve documents for certain queries that Elasticsearch cannot.

Model	Development		Evaluation	
	P@1	MRR	P@1	MRR
Elastic_{10}	0	16.07	0	16.99
Struct_w	14.44	23.57	19.68	28.11
Struct_l	14.81	**25.59**	**20.97**	**30.14**
Struct_e	**15.56**	24.47	20.32	29.22

Table 6: Results on queries failed by Elasticsearch.

To validate our hypothesis, we test structure matching on the subset of queries failed by Elasticsearch. We first take the top-10 results from Elasticsearch then rerank the results using the scores from structure matching for queries that Elasticsearch gives P@1 of 0%. As shown in Table 6, structure matching is capable of reranking a significant portion (around 20%) of these queries correctly, establishing that our hypothesis is true.

5.3 Document Reranking

The scores from Elastic_{10} and Struct_* for each document are fed into the binary classifier that decides whether or not to accept the top-1 result from Elasticsearch. If not, the documents are reranked by the weighted sum of these scores (Section 4.3). The Rerank_1 row in Table 4 shows the results when all the weights = 1, which gives an over 4% improvement of P@1 on the evaluation set. The Rerank_λ row shows the results when the optimized weights are used, which gives an additional 3% boost on the development set but not on the evaluation set.

It is worth mentioning that we initially tackled this as a document classification task using convolutional neural networks similar to Kim (2014); however, it gave P@1 $\approx$ 20% and MRR $\approx$ 33%. Such poor results were due to the huge size of our documents, over 4.6K words on average, beyond the capacity of a CNN. Thus, we decided to focus on reranking, which gave the best performance.

6 Conclusion

We propose a cross-genre document retrieval task that matches between TV show transcripts and their descriptions in summaries and plots. Our structure reranking approach gives an improvement of more than 4% of P@1, showing promising results for this task. In the future, we will add more structural information such as coreference relations to our structure matching and apply a more sophisticated parameter optimization technique such as the Bayesian optimization for finding λ_*.

[5] www.elastic.co/products/elasticsearch

References

Jaroslaw Baliński and Czeslaw Daniłowicz. 2005. Reranking method based on inter-document distances. *Information processing & management* 41(4):759–775.

David C Blair and Melvin E Maron. 1985. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM* 28(3):289–299.

James P Callan. 1994. Passage-level evidence in document retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., pages 302–310.

Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 186–193.

Yu-Hsin Chen and Jinho D. Choi. 2016. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Los Angeles, pages 90–100. http://www.aclweb.org/anthology/W16-3612.

Raquel Fernández, Staffan Larsson, Robin Cooper, Jonathan Ginzburg, and David Schlangen. 2011. Reciprocal learning via dialogue interaction: Challenges and prospects. In *Proceedings of the IJCAI 2011 Workshop on Agents Learning Interactively from Human Teachers (ALIHT 2011)*.

Annika Flycht-Eriksson and Arne Jönsson. 2003. Some empirical findings on dialogue management and domain ontologies in dialogue systems - implications from an evaluation of birdquest. In Akira Kurematsu, Alexander Rudnicky, and Syun Tutiya, editors, *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*. pages 158–167. http://www.aclweb.org/anthology/W03-2113.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 1066–1076. http://www.aclweb.org/anthology/N15-1113.

Zhiyang He, Xien Liu, Ping Lv, and Ji Wu. 2016. Hidden softmax sequence model for dialogue structure analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2063–2072. http://www.aclweb.org/anthology/P16-1194.

Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *HLT-NAACL*. pages 851–861.

Seokhwan Kim, Rafael Banchs, and Haizhou Li. 2016. Exploring convolutional and recurrent neural networks in sequential labelling for dialogue topic tracking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 963–973. http://www.aclweb.org/anthology/P16-1091.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Doha, Qatar, EMNLP'14, pages 1746–1751. http://www.aclweb.org/anthology/D14-1181.

Amita Misra, Pranav Anand, Jean E. Fox Tree, and Marilyn Walker. 2015. Using summarization to discover argument facets in online idealogical dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 430–440. http://www.aclweb.org/anthology/N15-1046.

Tatsuro Oya and Giuseppe Carenini. 2014. Extractive summarization and dialogue act modeling on email threads: An integrated probabilistic approach. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Association for Computational Linguistics, Philadelphia, PA, U.S.A., pages 133–140. http://www.aclweb.org/anthology/W14-4318.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval* 3(4):333–389.

Lu Wang and Claire Cardie. 2012. Focused meeting summarization via unsupervised relation extraction. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Seoul, South Korea, pages 304–313. http://www.aclweb.org/anthology/W12-1642.

Jason D Williams, Nobal B Niraula, Pradeep Dasigi, Aparna Lakshmiratan, Carlos Garcia Jurado Suarez, Mouni Reddy, and Geoff Zweig. 2015. Rapidly scaling dialog systems with interactive learning. In *Natural Language Dialog Systems and Intelligent Assistants*, Springer, pages 1–13.

Yang Xu and David Reitter. 2016. Entropy converges between dialogue participants: Explanations from an information-theoretic perspective. In *Proceedings of the 54th Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pages 537–546. http://www.aclweb.org/anthology/P16-1051.

Lingpeng Yang, Donghong Ji, Guodong Zhou, Yu Nie, and Guozheng Xiao. 2006. Document re-ranking using cluster validation and label propagation. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, pages 690–697.

Koichiro Yoshino, Shinsuke Mori, and Tatsuya Kawahara. 2011. Spoken dialogue system based on information extraction using similarity of predicate argument structures. In *Proceedings of the SIGDIAL 2011 Conference*. Association for Computational Linguistics, Portland, Oregon, pages 59–66. http://www.aclweb.org/anthology/W11-2008.

Dong Zhou and Vincent Wade. 2009. Latent document re-ranking. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, pages 1571–1580.