

# Multi-level Abstraction Convolutional Model with Weak Supervision for Information Retrieval

Yifan Nie  
Université de Montréal  
Montréal, Québec  
yifan.nie@umontreal.ca

Alessandro Sordoni  
Microsoft Research  
Montréal, Québec  
alsordon@microsoft.com

Jian-Yun Nie  
Université de Montréal  
Montréal, Québec  
nie@iro.umontreal.ca

## ABSTRACT

Recent neural models for IR have produced good retrieval effectiveness compared with traditional models. Yet all of them assume that a single matching function should be used for all queries. In practice, user's queries may be of various nature which might require different levels of matching, from low level word matching to high level conceptual matching. To cope with this problem, we propose a multi-level abstraction convolutional model (MACM) that generates and aggregates several levels of matching scores. Weak supervision is used to address the problem of large training data. Experimental results demonstrated the effectiveness of our proposed MACM model.

评测: ClueWeb

## CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*;

## KEYWORDS

Information Retrieval; Neural Network; Ranking

## ACM Reference Format:

Yifan Nie, Alessandro Sordoni, and Jian-Yun Nie. 2018. Multi-level Abstraction Convolutional Model with Weak Supervision for Information Retrieval. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, July 8–12, 2018, Ann Arbor, MI, USA. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3209978.3210123>

## 1 INTRODUCTION

Recently, several deep learning models for information retrieval have been proposed [2, 4, 6, 8]. These models have demonstrated their potential to improve the effectiveness in ad-hoc search. In general, a deep neural model is constructed to represent the content of the document and the query [4, 8], and/or their interactions or matching scores [2, 6]. The utilization of deep neural models is motivated by their ability to make high level and more abstract matching between the document and the query, thereby alleviating the vocabulary mismatch problem. We observe, however, that these models only use one level of final representation or matching score for any document-query pair. For example, in [8], several layers of convolutions are used to create more and more abstract

representations for the document and the query, and the matching score only relies on the last layer of representation. Convolution is an operation that aggregates lower-level features to produce more abstract features. A matching score at the highest level tends to reflect a conceptual matching. In reality, user's queries may be of different nature. Some queries such as "Ron Howard" (a query in ClueWeb) asking for information about a celebrity would require a low level lexical matching rather than conceptual matching. We call them lexical queries. On the other hand, a query like "lymphoma in dogs" is intended to find document about corresponding concept(s), therefore a high level conceptual matching is preferred. These queries are called conceptual queries. These examples clearly show the need for matching document and query at different levels of abstraction. Inspired by this intuition, in this paper we propose a Multi-level Abstraction Convolutional Model (MACM), which integrates document-query matching at different levels of abstraction. This model is expected to have a better capability of coping with different types of user queries.

Although neural IR models can focus either on document and query representation or on interactions between them, Guo et al. [2] showed that the latter is more effective than the former. Based on this observation, our model is built on document-query interactions rather than representations. A critical problem in building deep neural models for IR is the requirement of a large amount of labeled training data, which is often unavailable. The idea of weak supervision by a traditional IR model is proposed recently [1] and shown to be effective. Inspired by this work, we employ the BM25 retrieval model [7] for weak supervision - the ranked documents retrieved by BM25 are used to train our deep neural model. We will see that this strategy is able to train our deep neural model, leading to superior effectiveness to BM25.

The main contribution of our paper lies in a new neural model capable of coping with different types of queries by matching them with documents at different levels of abstraction. This idea can be easily adopted in other deep neural models, whether they are based on representations or interactions, use CNN or RNN. Our experiments on ClueWeb confirm that our approach can result in superior retrieval effectiveness.

## 2 RELATED WORK

The core matching mechanism is responsible to produce a relevance score  $rel$  between the query  $q$ , and the document  $d$ . Depending on how  $rel$  is calculated, previous deep IR models could be roughly categorized into representation-based models and interaction-based models. In representation-based model, the relevance score could

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions.acm.org](https://permissions.acm.org).

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5657-2/18/07...\$15.00

<https://doi.org/10.1145/3209978.3210123>

be calculated by Equation 1.

$$rel = S(\phi(q), \phi(d)) \quad (1)$$

where  $\phi$  is a complex feature function to map query or document text into meaningful semantic representations through several hidden layers.  $S$  is a matching function, usually cosine or dot similarity, which maps representations to matching scores. For example, in DSSM [4],  $\phi$  is a feed forward neural network and  $S$  is a simple cosine similarity. In CDSSM [8],  $\phi$  is a convolutional network and  $S$  is the cosine similarity.

In interaction-based models such as MatchPyramid [6] and ARC-II [3], the relevance scores are calculated in a different way as depicted in Equation 2.

$$rel = S_n \circ S_{n-1} \circ \dots \circ S_0(w(q), w(d)) \quad (2)$$

where the feature function  $w$  is often a simple embedding look-up function which maps the term into its word embedding vector, and the matching function is a composition of a series of neural layers  $S_0, S_1, \dots, S_n$ .

In a similar way, the DRMM model [2] generates the histogram of interaction intensities between each query term  $t_i^q$  and all document terms  $t_j^d$  as input features. Then  $n$  shared-weight feed-forward neural networks are built to learn the underlying matching patterns between this query term  $t_i^q$  and the whole document  $d$  and output relevance scores  $s_i$ , where  $n$  is the query length. Finally a gating mechanism aggregates the  $n$  relevance scores to produce an overall matching score  $S$ . Guo et al. [2] showed that the interaction-based model can outperform the representation-based model for ad-hoc retrieval, and it could better capture match signals and query term importance. In our study, we will focus on interaction-based models.

We notice that all the above models use only the final level of representation or interaction pattern, ignoring all intermediate ones in the networks. As a result, any query is matched with documents at the same abstraction level. As we stated earlier, the user's query may be of different nature, which might require to match documents at different levels of abstraction. Therefore a natural question is whether the different levels of representation or interaction can be leveraged to cope with the needs of different queries. This is the question we address in this paper.

### 3 MULTI-LEVEL ABSTRACTION CONVOLUTIONAL MODEL

In order to investigate the effectiveness of exploiting the matching scores of multiple abstraction levels, in this study a Multi-level Abstraction Convolutional Model (MACM) is proposed. The architecture is illustrated in Fig 1, which contains several levels of interaction matrices, resulting in different matching scores.

**Convolutional Architecture:** In this architecture, the query and document are represented by a set of word embedding vectors  $q = [t_1^{(q)}, \dots, t_n^{(q)}]$ , and  $d = [t_1^{(d)}, \dots, t_m^{(d)}]$ , where  $t_i^{(q)}$  and  $t_j^{(d)}$  represent the embedding vectors for query term  $i$  and document term  $j$  respectively, and  $n, m$  are the query length and document length respectively. Following [6], an interaction matrix  $I$  is constructed as follows.

$$I_{ij} = \cos(t_i^{(q)}, t_j^{(d)}) \quad (3)$$

基于 word embedding 的交互矩阵。

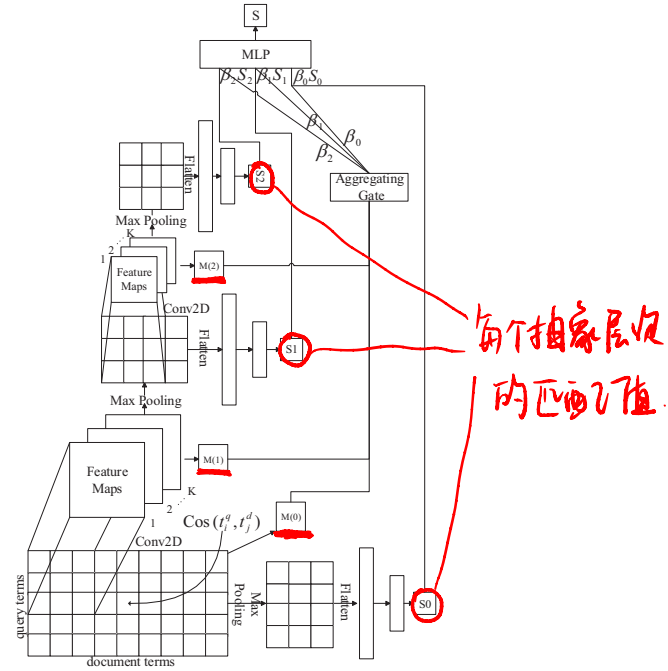


Figure 1: Multi-level Abstraction Convolutional Model

Once the input interaction matrix is constructed, a series of convolution and max-pooling layers are added on top in order to build more abstract interaction patterns.

$$P_0 = \max\_pool(I) \quad (4)$$

$$C_1^k = f(W_1^k * I + b_1^k), k = 1, \dots, K \quad (5)$$

$$P_1^k = \max\_pool(C_1^k), k = 1, \dots, K \quad (6)$$

$$C_i^k = f(W_i^k * P_{i-1}^k + b_i^k), i = 2, \dots, L, k = 1, \dots, K \quad (7)$$

$$P_i^k = \max\_pool(C_i^k), i = 2, \dots, L, k = 1, \dots, K \quad (8)$$

where  $C_i^k$  is the feature map  $k$  of the  $i^{th}$  convolved layer;  $I$  is the input interaction matrix;  $W_i^k$  and  $b_i^k$  are the kernel and bias of layer  $i$  for the feature map  $k$ ;  $L$  is the number of convolution layers, and  $K$  is the number of feature maps;  $f$  is a non-linear mapping; and  $*$  represents the convolution operator.

In order to determine the matching scores of each abstraction level, each max-pooled layer  $P_i$ ,  $i = 0, \dots, L$  is flattened into a 1D vector and fed into a fully connected MLP to output a scalar score  $S_i$  at this level.

$$h_i = g(W_i^h P_i + b_i^h), i = 0, \dots, L \quad (9)$$

$$S_i = h(W_i^s h_i + b_i^s), i = 0, \dots, L \quad (10)$$

where  $h_i$  and  $S_i$  represent the hidden layer of the MLP and the matching score for the  $i^{th}$  convolution layer respectively;  $W_i^h, b_i^h, W_i^s, b_i^s$  are the weights and biases for the hidden and scoring layer;  $g$  and  $h$  are non-linear mappings.

**Matching Score Aggregation:** In order to combine the matching scores of different abstraction levels  $S_i$ , in this architecture, we

把每个抽象层次的结果  $P_i$  进行 flatten + MLP, 得到匹配值。

propose a gating mechanism to aggregate the scores  $S_i$  by considering the importance of each abstraction level. A possible implementation of the gating mechanism is to make it dependent on the nature (class) of the query. An alternative way is to make it dependent on the matching scores at different levels, based on the intuition that a strong matching score at a level means that this level is appropriate. We use the second strategy in this paper.

For the input interaction matrix  $I$  and each convolved layer  $C_i$ , a scalar feature  $M^{(i)}$  will be calculated. For the input interaction matrix  $I$ , we take the max interaction values  $M_u^{(0)}$  across each row  $u$ , which represents the max matching intensity across all document terms for the query term  $t_u^{(q)}$ . Afterwards, we sum up all the  $M_u^{(0)}$ s for each query term  $t_u^{(q)}$  and get the global maximum interaction value  $M^{(0)}$  for the whole query with respect to the document. This quantity reflects the word-level matching between document and query. The process could be summarized as follows.

计算  $M^{(0)}$ : 
$$M_u^{(0)} = \max_{v=1..m} I_{uv}, \quad M^{(0)} = \sum_{u=1}^n M_u^{(0)} \quad (11)$$

For each feature map  $C_i^{(k)}$  in the convolved layer  $i$ , we proceed in the same way to obtain a  $M_{(k)}^{(i)}$  for this specific feature map  $k$ . Then we average the  $M_{(k)}^{(i)}$ s to obtain the global  $M^{(i)}$  for this convolution layer. The process could be summarized as follows.

计算  $M^{(i)}$ : 
$$M_{u,(k)}^{(i)} = \max_{v=1..m} [C_i^{(k)}]_{uv}, \quad u = 1, \dots, n, \quad k = 1, \dots, K \quad (12)$$

$$M_{(k)}^{(i)} = \sum_{u=1}^n M_{u,(k)}^{(i)}, \quad M^{(i)} = \frac{1}{K} \sum_{k=1}^K M_{(k)}^{(i)} \quad (13)$$

Finally, the  $M$  values are normalized through a softmax gate as follows.

根据  $M$  计算权重: 
$$\beta_i = \frac{\exp(\alpha_i M^{(i)})}{\exp(\sum_{j=0}^L \alpha_j M^{(j)})} \quad (14)$$

where  $\alpha_i$  are learnable parameters,  $M^{(i)}$  are the  $M$  values for each convolution layer  $i$ , and  $L$  is the total number of convolution layers.

The interaction scores  $S_i$  are then weighted and concatenated as  $[\beta_0 S_0, \dots, \beta_L S_L]$  to be fed into a MLP aggregator to obtain an overall relevance score.

计算最终相似值: 
$$S = f(W[\beta_0 S_0, \dots, \beta_L S_L] + b) \quad (15)$$

The training is done with a pair-wise loss: given a training example  $(Q, D_+, D_-)$ , we hope that the score  $S(Q, D_+)$  should be higher than the score  $S(Q, D_-)$ . The loss is defined in Equation 16, where  $\Theta$  includes all trainable parameters in this MACM architecture.

loss: 
$$L(Q, D_+, D_-; \Theta) = \max(0, 1 - (S(Q, D_+) - S(Q, D_-))) \quad (16)$$

## 4 EXPERIMENTAL STUDY

### 4.1 Dataset and Settings

Experiments are conducted on the ClueWeb09B collection. The detailed statistics of the dataset are summarized in Table 1. To generate weak supervision labels, we employ the AOL query logs<sup>1</sup> and filter out navigational queries<sup>2</sup> and queries containing non-alphanumeric

<sup>1</sup><http://octopus.inf.utfsml.cl/~juan/datasets/>

<sup>2</sup>Queries containing URL strings ("www.", ".com", ".org", ".net", ".edu")

Table 1: Collection Statistics

Collection	Genre	Validation Queries	Test Queries	#Docs	Avg.d.length
Clueweb09B	Webpages	1-50	51-200	50M	1,506

characters as done in [1]. This results in 8,969,337 training queries. We retrieve the top 50 documents using Indri<sup>3</sup> BM25 model with default parameters ( $k_1 = 1.2, b = 0.75, k_3 = 1000$ ). For each validation and test query, we return top 1000 documents by BM25 model as candidate documents and use our model to rerank them by the inferred matching score  $S$ .

During training, for a given query, we randomly sample 2 documents and regard the one with higher BM25 score as positive document, the other one as negative document. We limit the max number of convolution layers  $L$  to be 2 and fix the number of feature maps to [32, 16] for the 2 convolution layers, set the max query length and document length to be  $n = 15, m = 1000$  and apply zero paddings as done in [6]. During training we only use queries whose length is no more than 15 and we checked that this limit is sufficient to deal with the queries in validation set and test set whose maximum query lengths are 4 and 5 respectively. We fix the pooling size of all max pooling layers to be (2, 2), employ fixed pretrained GloVe.6B.300d embeddings<sup>4</sup>, and omit OOV document terms. Other alternative settings will be explored in our future work.

### 4.2 Results and Discussions

We employ the mean average precision (MAP) and nDCG [5] as evaluation metrics. The main experimental results are summarized in the first section of Table 2. We conducted paired t-test against BM25, and the statistically significant results ( $p < 0.05$ ) are marked with \*. The MACM model has one column ( $col = 1$ ) of 2-layered convolutions ( $L = 2$ ) and the filter shapes are set to (3, 3) and (5, 5) for the first and second convolution layer, which roughly correspond to phrase and sentence matching. We do not increase further the filter size because of the short length of queries.

We first observe that our MACM model trained with weak supervision by BM25 can outperform BM25. This observation is consistent with that of [1]. This confirms that a deep neural model has a higher generalizability than BM25. In order to test the usefulness of leveraging different levels of matching, we also build *BaseModel* -  $S_i$  where only the  $i^{th}$  level matching score  $S_i$  is used as the overall matching score. This corresponds to the strategy generally used in the current deep IR models. The results show that our MACM model outperforms all the *BaseModel* -  $S_i$  on all the evaluation measures. This result confirms the usefulness of multi-level matching. To further analyze the experimental results, we compare our model with some base models on some queries in Table 3.

For queries which ask for an exact or near-exact match, such as "Porterville" (a city in California) and "Ron Howard" (an actor), the model with only term-level  $S_0$  matching outperforms the model with high-level score  $S_2$ . The latter model may expand too much the semantic of the query, which performs poorly. For example among the documents retrieved for the query "Ron Howard" by

<sup>3</sup><https://www.lemurproject.org/indri.php>

<sup>4</sup><https://nlp.stanford.edu/projects/glove/>

**Table 2: Experimental Results**

Model	MAP	NDCG@1	NDCG@3	NDCG@10	NDCG@20
BM25	0.0879	0.1178	0.1359	0.1356	0.1394
MACM	<b>0.0928</b>	<b>0.1610*</b>	<b>0.1483</b>	<b>0.1431</b>	<b>0.1424</b>
Basemodel-S0	0.0546*	0.1218	0.1020*	0.0991*	0.0938*
Basemodel-S1	0.0789*	0.1218	0.1254	0.1283	0.1272*
Basemodel-S2	0.0884	0.1485*	0.1423	0.1411	0.1389
MACM-1col	<b>0.0928</b>	<b>0.1610</b>	<b>0.1483</b>	<b>0.1431</b>	<b>0.1424</b>
MACM-2cols	0.0894	0.1357	0.1381	0.1402	0.1409
MACM-3cols	0.0817	0.1352	0.1337	0.1286	0.1293
MACM-1col-(1,3)(1,5)	0.0810	0.1066	0.1202	0.1220	0.1262
MACM-1col-(3,3)(1,5)	0.0892	0.1451	0.1480	0.1415	0.1483
MACM-1col-(3,3)(5,5)	<b>0.0928</b>	<b>0.1610</b>	<b>0.1483</b>	0.1431	0.1424
MACM-1col-(3,3)(1,10)	0.0921	0.1519	0.1465	0.1442	0.1461

**Table 3: nDCG@10 of Representative Queries**

Topic_num	Query	BaseModel-S0	BaseModel-S2	MACM
159	Porterville	0.75109	0.40434	<b>0.81458</b>
171	Ron Howard	0.01997	0.01324	<b>0.02683</b>
111	lymphoma in dogs	0.01938	0.16179	<b>0.27686</b>
192	condos in Florida	0.13886	0.18873	<b>0.31046</b>

*BaseModel* – S2, a number of documents are about other people named “Howard” which are semantically related to “Ron Howard” but are irrelevant for this query. This is an example of over generalization. For conceptual queries such as “lymphoma in dogs”, “condos in Florida”, the model with high-level score  $S_2$  outperforms the model with only term-level score  $S_0$ , although the *BaseModel*-S0 still yields quite good result. This query is an example that requires some abstraction. The “lymphoma in dogs” is a typical example of conceptual query which requires even more abstraction. We can observe a large difference between *BaseModel*-S0 and *BaseModel*-S2 for this query. Indeed, the documents retrieved for “lymphoma in dogs” by *BaseModel* – S2 include the desired documents about “veterinary cancer treatment”, “pet chemotherapy”. In all the above cases, by dynamically combining the matching scores of the 3 levels of abstraction by a gating mechanism, our MACM model can outperform the single-level base models. These examples show the ability of our model to use the appropriate level(s) of matching depending on the query.

### 4.3 Alternative Configurations of the Networks

We study now the influence of the number of parallel convolutions with different granularities. The parallel convolutions aim to capture different types of features, which have also been used in text matching models [9].

The results are presented in the 2nd section of Table 2. The MACM-1col model has only one column of 2 convolution layers with filter sizes  $FL_1 = (3, 3)$ ,  $FL_2 = (5, 5)$  as before. The MACM-2cols and MACM-3cols have 2 and 3 parallel columns of 2-layered convolutions, with filter sizes set to be  $FL_1 = [(3, 3), (5, 5)]$ ,  $FL_2 = [(5, 5), (6, 10)]$  and  $FL_1 = [(3, 3), (5, 5), (7, 7)]$ ,  $FL_2 = [(5, 5), (6, 10), (7, 50)]$ .  $FL_i$  stands for the filter sizes in the convolution layer  $i$ . However, experimental results show that multiple columns of convolution do not help. One possible explanation is that in the one

column model, the 2nd convolution layer can already capture the information of a larger granularity than the 1st convolution layer and the extra columns of convolution with larger filters play a similar role.

We also tested different shapes of filters. The results are presented in the 3rd section of Table 2. We build 1-column MACM model with different filter shapes for the 1st and 2nd convolution layers. From the experiment results, we find that squared filters ( $n \times n$ ) work better than rectangular filters ( $m \times n$ ). Intuitively, squared filters can possibly capture the matching signals of  $n$ -grams, whereas the rectangular filters fail to model the  $n$ -gram matching patterns. More investigations are needed to better understand the reasons.

## 5 CONCLUSION AND FUTURE WORK

The existing deep neural models for IR create a matching score at the same level of abstraction for any query. They may fail to cope with queries of different nature. In this paper, to address this issue, we proposed a Multi-level Abstraction Convolution Model trained under weak supervision. Experimental results showed that our proposed MACM could outperform the BM25 baseline and demonstrated the usefulness of multi-level matching. This work can be extended in several directions. First, the impact of different shapes of convolution filter requires further investigation. Second, it is possible to extend it to representation-based deep models as well. Finally, the idea could also be used in RNN-based models.

## ACKNOWLEDGEMENT

This research is partly supported by a NSERC discovery grant and a Quebec-China PhD scholarship.

## REFERENCES

- [1] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 65–74.
- [2] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*. 55–64.
- [3] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2042–2050.
- [4] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using click-through data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*. 2333–2338.
- [5] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.
- [6] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2016. A Study of MatchPyramid Models on Ad-hoc Retrieval. *CoRR* abs/1606.04648 (2016).
- [7] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.
- [8] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*. 373–374.
- [9] Wenpeng Yin and Hinrich Schütze. 2015. MultiGranCNN: An Architecture for General Matching of Text Chunks on Multiple Levels of Granularity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. 63–73.