UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Neural Networks for Information Retrieval

Kenter, T.; Borisov, A.; Van Gysel, C.; Dehghani, M.; de Rijke, M.; Mitra, B.

DOI
10.1145/3077136.3082062

Publication date
2017

Document Version
Author accepted manuscript

Published in
SIGIR'17 : proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval

Link to publication

Citation for published version (APA):
Kenter, T., Borisov, A., Van Gysel, C., Dehghani, M., de Rijke, M., & Mitra, B. (2017). Neural Networks for Information Retrieval. In *SIGIR'17 : proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval: August 7-11, 2017, Shinjuku, Tokyo, Japan* (pp. 1403-1406). Association for Computing Machinery. https://doi.org/10.1145/3077136.3082062

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:13 Oct 2026

Neural Networks for Information Retrieval

Tom Kenter*
University of Amsterdam
Amsterdam, The Netherlands
tom.kenter@uva.nl

Alexey Borisov
Yandex
Moscow, Russia
alborisov@yandex-team.ru

Christophe Van Gysel
University of Amsterdam
Amsterdam, The Netherlands
cvangysel@uva.nl

Mostafa Dehghani
University of Amsterdam
Amsterdam, The Netherlands
dehghani@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

Bhaskar Mitra
Microsoft, University College London
Cambridge, UK
bmitra@microsoft.com

arXiv:1707.04242v1 [cs.IR] 13 Jul 2017

CCS CONCEPTS

• **Information systems** → *Personalization*; *Probabilistic retrieval models*; *Language models*; *Learning to rank*; *Question answering*;

EXTENDED ABSTRACT

Machine learning plays a role in many aspects of modern IR systems, and deep learning is applied in all of them. The fast pace of modern-day research has given rise to many different approaches for many different IR problems. The amount of information available can be overwhelming both for junior students and for experienced researchers looking for new research topics and directions. Additionally, it is interesting to see what key insights into IR problems the new technologies are able to give us. The aim of this full-day tutorial is to give a clear overview of current tried-and-trusted neural methods in IR and how they benefit IR research. It covers key architectures, as well as the most promising future directions.

1 MOTIVATION

Prompted by the advances of deep learning in computer vision research, neural networks have resurfaced as a popular machine learning paradigm in many other directions of research as well, including information retrieval. Recent years have seen neural networks being applied to all key parts of the typical modern IR pipeline, such core ranking algorithms [26, 42, 51], click models [9, 10], knowledge graphs [8, 35], text similarity [28, 47], entity retrieval [52, 53], language modeling [5], question answering [22, 56], and dialogue systems [34, 54].

A key advantage that sets neural networks apart from many learning strategies employed earlier, is their ability to work from raw input data. E.g., when given enough training data, well-designed networks can become feature extractors themselves, e.g., incorporating basic input characteristics such as term frequency (tf) and term saliency (idf)—that used to be pre-calculated offline—in their initial layers. Where designing features used to be a crucial aspect and contribution of newly proposed IR approaches, the focus has shifted to designing network architectures instead. As a consequence, many different architectures and paradigms have been proposed, such as auto-encoders, recursive networks, recurrent networks, convolutional networks, various embedding methods, deep reinforcement and deep q-learning, and, more recently, generative adversarial networks, of which most have been applied in IR settings. The aim of the *neural networks for IR (NN4IR)* tutorial is to provide a clear overview of the main network architectures currently applied in IR and to show explicitly how they relate to previous work. The tutorial covers methods applied in industry and academia, with in-depth insights into the underlying theory, core IR tasks, applicability, key assets and handicaps, scalability concerns and practical tips and tricks.

We expect the tutorial to be useful both for academic and industrial researchers and practitioners who either want to develop new neural models, use them in their own research in other areas or apply the models described here to improve actual IR systems.

2 OBJECTIVES

The material in the tutorial covers a broad range of IR applications. It is structured as follows:

Preliminaries *(60 minutes).* The recent surge of interest in deep learning has given rise to a myriad of architectures. Different though the inner structures of neural networks can be, there are many concepts common to all of them. This first session covers the preliminaries; we briefly recapitulate the basic concepts involved in neural systems, such as back propagation [44], distributed representations/embeddings [39], convolutional layers [30], recurrent networks [38], sequence-to-sequence models [50], dropout [49], loss functions, optimization schemes like Adam [29].

Semantic matching I: supervised learning *(60 minutes).* The problem of matching items based on their textual descriptions arises in many IR systems. The traditional approach involves counting query term occurrences in the description text (e.g., BM25 [43]). However, to bridge the lexical gap caused by vocabulary-related and linguistic differences many latent semantic models have been proposed [6, 13, 24, 55], and more recently neural embedding methods [39]. In this session we will focus on semantic matching settings where a supervised signal is available. The signal can be explicit, such as a label for learning task-specific latent representations [25–27, 36, 47, 48], or relevance labels and, more implicitly, clicks for neural IR methods [15, 19, 31, 41, 42].

*Corresponding author.

SIGIR '17, Shinjuku, Tokyo, Japan
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of SIGIR '17, August 07-11, 2017*, http://dx.doi.org/10.1145/3077136.3082062.

Semantic matching II: Semi- and unsupervised learning *(60 minutes).* How to learn semantics in the absence of relevance labels or user interaction signals? Depending on the available resources, one can choose for semi- or unsupervised matching models.

Unsupervised semantic matching methods can be categorized into two groups. First, *using pre-trained word embeddings* like combining traditional retrieval models with an embedding-based translation model [16, 58], using pre-trained embeddings for query expansion to improve retrieval [57], and representing documents as Bag-of-Word-Embeddings (BoWE) [20, 27]. Second, *learning representations from scratch* like learning representations of words and documents [28, 32] and employing them in retrieval task [2, 3], and learning representations in an end-to-end neural model for learning a specific task like entity ranking for expert finding [53] or product search [52].

In semi-supervised learning, on the other hand, queries (without relevance labels), or prior knowledge about document similarity can be used to induce pseudo-relevance labels. Furthermore, it is possible to use heuristic methods to generate weak supervision signals and to go beyond them by employing proper learning objectives and network designs [14].

Learning to rank *(45 minutes).* Capturing the notion of relevance for ranking needs to account for different aspects of the query, the document, and their relationship. Neural methods for ranking can use both manually crafted features from query and document and combine them with regards to a ranking objective, or learn latent representations for them in situ.

Irrespective of how the query and the documents are featurized, a neural learning to rank model can be designed for different scenarios, each having its own appropriate loss function. An example is the point-wise versus pair-wise paradigm, each of which has a different objective that calibrates either scores or the relative ranking of documents, given a query. Neural learning to rank models can also be designed to be provided with different levels of supervision during training—unsupervised [45, 52, 53], semi/weakly-supervised [14, 51], or fully-supervised using labeled [42] or click data [26].

Modeling user behavior *(45 minutes).* Modeling user browsing behavior plays an important role in the development of modern IR systems. Accurately interpreting user clicks is difficult due to various types of bias. For example, users tend to click more on results ranked on top positions (position bias) and visually salient results (attention bias). The traditional way to account for these biases is to design a Probabilistic Graphical Model (PGM) that explains relationships between click/skip events (observed variables) and examination (unobserved variables). Over the last decade many PGM-based click models have been proposed (see [12] for an overview). However, these click models can model only those patterns that are explicitly encoded in their PGMs. Recently, it was shown that recurrent neural networks can learn to account for biases in user clicks directly from the click-through data, i.e., without the need for a predefined set of rules as is customary for PGM-based click models [9]. Additionally, there are similar biases in click dwell times, which the neural approach can account for too.

Generating responses *(45 minutes).* Recent inventions such as smart home devices, voice search and virtual assistants provide new ways of accessing information. They require a different response format than the classic ten blue links. Targeting this newly emerging demand, some models have been proposed to respond by generating natural language replies on the fly, rather than by (re)ranking a fixed set of items or extracting passages from existing pages.

Examples are conversational and dialog systems [7, 34, 54] or machine reading and question answering tasks where the model either infers the answer from unstructured data, like textual documents that do not necessarily feature the answer literally [21, 22, 46, 56], or generates natural language given structured data, like data from knowledge graphs or from external memories [1, 18, 33, 37, 40].

Outlook *(30 minutes).* In this session, open research questions and future directions are discussed. One of the big challenges for IR at the moment is how to process full document text using neural networks. On a higher level, it is probably desirable to learn all components of a full IR system in an end-to-end fashion.

Another challenge is maintaining long term (multiple day) search sessions or conversations. Which naturally leads to an additional open problem: how to evaluate (neural) conversational systems.

Finally, we cover recent advances, like Generative Adversarial Networks [17].

Summing up, the **objectives of the NN4IR tutorial** are as follows:

- Give an extensive overview of neural network architectures currently employed in IR, both in academia and industry.
- Provide theoretical background, thereby equipping participants with the necessary means to form intuitions about various neural methods and their applicability.
- Identify the IR lessons learned by employing neural methods.
- Give practical tips and tricks, regarding network design, optimization, hyperparameter values, based on industry best practice.
- Discuss promising future research directions.

The **target audience** consists of researchers and developers in information retrieval who are interested in gaining an in-depth understanding of neural models across a wide range of IR problems. The tutorial will be useful as an overview for anyone new to the deep learning field as well as for practitioners seeking concrete recipes. The tutorial aims to provide a map of the increasingly rich landscape of neural models in IR.

By the end of the tutorial, attendees will be familiar with the main architectures of neural networks as applied in IR and they will have informed intuitions of their key properties and of the insights they bring into core IR problems. We aim to provide an overview of the main directions currently employed, together with a clear understanding of the underlying theory and insights, illustrated with examples.

3 FORMAT AND DETAILED SCHEDULE

Table 1 gives an overview of the time schedule of the tutorial. Below we provide the details for each session.

We bring a team team of six lecturers, all with their specific areas of specialization. Each session will have two expert lecturers who will together present the session. The initials below refer to the lecturers for this tutorial.

Preliminaries — 60 minutes (TK, MdR).

Table 1: Time schedule for NN4IR tutorial

Preliminaries	60 minutes
Semantic Matching I	60 minutes
Semantic Matching II	60 minutes
lunch break	
Learning to Rank	45 minutes
Modeling User Behavior	45 minutes
Generating Responses	45 minutes
Outlook	30 minutes
Wrap up	15 minutes

10 mins Back propagation – Given a standard feedforward network, we show the math and the intuition of back propagation. Also we will briefly touch on dropout.

5 mins Distributed representations – We show what a distributed representation is, and how distributed representations can be used.

10 mins Recurrent neural networks – We cover the basics, based on a language modeling scenario, including LSTMs [23] and GRUs [11].

10 mins Embedding methods – We detail how word2vec works and how it can be applied to different settings.

10 mins Sequence-to-sequence models – Basic architecture of seq2seq models [50], including the attention mechanism [4].

10 mins Convolutional networks – CNNs are primarily employed in computer vision, but can be beneficial in text classification tasks too.

5 mins Optimisation schemes – Standard back propagation with a fixed learning rate is typically replaced by more sophisticated schemes that handle learning rate annealing.

Semantic matching I: supervised — 60 minutes (AB, BM).

10 mins Short text similarity – Given two short texts, e.g., queries or sentences, how can we predict if they are semantically similar?

15 mins Word embeddings for matching – Learning embeddings from click data.

20 mins Deep neural architectures for matching – Deep Structured Semantic Model (DSSM) [26].

15 mins Learning to match using local representations – Use both local and global representations for query-document matching.

Semantic matching II: Semi- and unsupervised semantic matching — 60 minutes (MD, CVG).

10 mins Semi-supervised semantic matching – We cover how to model pseudo-labeling using prior knowledge like document similarity, or by employing heuristic methods as weak supervision signals.

25 mins Unsupervised semantic matching using pre-trained word embeddings – We show how different IR tasks benefit from using pre-trained word embeddings by pre-estimating representations for query and documents, or as warm start for representation learning during training.

25 mins Learning unsupervised representations from scratch for semantic matching – We explain how to learn representations of words and documents in an unsupervised manner without any relevance label, that satisfy the requirements of IR problems.

Learning to rank, 45 minutes (CVG, BM).

10 mins Feature-based models for representation learning – We explain how to train a ranker using featurized input, and how to feed the network with raw data to have it learn representations jointly with a downstream task.

15 mins Ranking objectives and loss functions – We describe pointwise and pair-wise settings for the ranking task and the proper loss functions for each setting.

20 mins Training under different levels of supervision – We cover how to train a neural ranker in an unsupervised way, with weak, semi- or full supervision and discuss requirements and concerns of each situation.

Modeling user behavior — 45 minutes (AB, MdR).

10 mins Biases and PGM-based click models – We introduce notions of bias in user behavior and explain how to account for them using probabilistic graphical models (PGMs).

25 mins Neural click models – We discuss weaknesses of PGM-based approaches and present an alternative based on recurrent neural networks.

10 mins Hybrid approach – We describe recent work on modeling biases in times between user actions (e.g., click dwell time) using ideas exploited in PGM-based and neural click models.

Generating responses — 45 minutes (MD, TK).

15 mins Machine reading/question answering – How is the sequence-to-sequence paradigm applied in neural QA systems.

15 mins Conversational IR/dialogue systems – Unlike QA systems, conversational systems should maintain a state of a session.

15 mins General chatbots – Chatbots bring their own set of challenges. How to stay consistent throughout the course of a conversation? How to maintain a persona?

Outlook — 30 minutes (all).

15 mins Recent advances

15 mins Open research questions, current challenges

Wrap up — 15 minutes (TK).

15 mins Overview of material presented and conclusion

4 TYPE OF SUPPORT MATERIALS TO BE SUPPLIED TO ATTENDEES

Slides Slides will be made publicly available on http://nn4ir.com.

Bibliography An annotated compilation of references will list all work discussed in the tutorial and should provide a good basis for further study.

Code Apart from the various open source neural toolkits (Tensorflow, Theano, Torch) many of the methods presented come with implementations released under an open source license. These will be discussed as part of the presentation of the models and algorithms. We provide a list pointers to available code bases.

REFERENCES

[1] S. Ahn, H. Choi, T. Pärnamaa, and Y. Bengio. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*, 2016.

[2] Q. Ai, L. Yang, J. Guo, and W. B. Croft. Analysis of the paragraph vector model for information retrieval. In *ICTIR*, pages 133–142. ACM, 2016.

[3] Q. Ai, L. Yang, J. Guo, and W. B. Croft. Improving language estimation with the paragraph vector model for ad-hoc retrieval. In *SIGIR*, pages 869–872. ACM, 2016.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2014.

[5] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[7] A. Bordes and J. Weston. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*, 2016.

[8] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *Conference on artificial intelligence*, 2011.

[9] A. Borisov, I. Markov, M. de Rijke, and P. Serdyukov. A neural click model for web search. In *WWW*, pages 531–541. International World Wide Web Conferences Steering Committee, 2016.

[10] A. Borisov, I. Markov, M. de Rijke, and P. Serdyukov. A context-aware time model for web search. In *SIGIR*, pages 205–214. ACM, 2016.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[12] A. Chuklin, I. Markov, and M. de Rijke. *Click Models for Web Search*. Morgan & Claypool, 2015.

[13] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[14] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft. Neural ranking models with weak supervision. In *Proc. SIGIR*, 2017.

[15] F. Diaz, B. Mitra, and N. Craswell. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*, 2016.

[16] D. Ganguly, D. Roy, M. Mitra, and G. J. Jones. Word embedding based generalized language model for information retrieval. In *SIGIR*, pages 795–798. ACM, 2015.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[18] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[19] M. Grbovic, N. Djuric, V. Radosavljevic, F. Silvestri, and N. Bhamidipati. Context- and content-aware embeddings for query rewriting in sponsored search. In *SIGIR*, pages 383–392. ACM, 2015.

[20] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. Semantic matching by non-linear word transportation for information retrieval. In *CIKM*, pages 701–710, New York, NY, USA, 2016. ACM.

[21] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701, 2015.

[22] D. Hewlett, A. Lacoste, L. Jones, I. Polosukhin, A. Fandrianto, J. Han, M. Kelcey, and D. Berthelot. WIKIREADING: A novel large-scale language understanding task over Wikipedia. In *ACL*, 2016.

[23] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[24] T. Hofmann. Probabilistic latent semantic indexing. In *Proc. SIGIR*, pages 50–57. ACM, 1999.

[25] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, pages 2042–2050, 2014.

[26] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proc. CIKM*, pages 2333–2338. ACM, 2013.

[27] T. Kenter and M. de Rijke. Short text similarity with word embeddings. In *CIKM*, pages 1411–1420. ACM, 2015.

[28] T. Kenter, A. Borisov, and M. de Rijke. Siamese cbow: Optimizing word embeddings for sentence representations. In *ACL*. ACL, 2016.

[29] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[31] M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger, et al. From word embeddings to document distances. In *ICML*, volume 15, pages 957–966, 2015.

[32] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[33] R. Lebret, D. Grangier, and M. Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.

[34] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. Deep reinforcement learning for dialogue generation. In *EMNLP*, 2016.

[35] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, 2015.

[36] Z. Lu and H. Li. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*, pages 1367–1375, 2013.

[37] H. Mei, M. Bansal, and M. R. Walter. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838*, 2015.

[38] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[40] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. In *EMNLP*, 2016.

[41] B. Mitra, E. Nalisnick, N. Craswell, and R. Caruana. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137*, 2016.

[42] B. Mitra, F. Diaz, and N. Craswell. Learning to match using local and distributed representations of text for web search. In *WWW '17*, 2017.

[43] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3), 1988.

[45] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[46] I. V. Serban, A. García-Durán, C. Gulcehre, S. Ahn, S. Chandar, A. Courville, and Y. Bengio. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv preprint arXiv:1603.06807*, 2016.

[47] A. Severyn and A. Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, pages 373–382. ACM, 2015.

[48] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM*, pages 101–110. ACM, 2014.

[49] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

[50] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[51] M. Szummer and E. Yilmaz. Semi-supervised learning to rank with preference regularization. In *CIKM*, pages 269–278. ACM, 2011.

[52] C. Van Gysel, M. de Rijke, and E. Kanoulas. Learning latent vector spaces for product search. In *CIKM '16*, pages 165–174, 2016.

[53] C. Van Gysel, M. de Rijke, and M. Worring. Unsupervised, efficient and semantic expertise retrieval. In *WWW '16*, pages 1069–1079, 2016.

[54] O. Vinyals and Q. Le. A neural conversational model. In *ICML*, 2015.

[55] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proc. SIGIR*, pages 178–185. ACM, 2006.

[56] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. In *ICLR*, 2016.

[57] H. Zamani and W. B. Croft. Estimating embedding vectors for queries. In *ICTIR*, pages 123–132. ACM, 2016.

[58] G. Zuccon, B. Koopman, P. Bruza, and L. Azzopardi. Integrating and evaluating neural word embeddings in information retrieval. In *ADCS*. ACM, 2015.