

Affect Enriched Word Embeddings for News Information Retrieval

Tommaso Teofili
Adobe
teofili@adobe.com

Niyati Chhaya
Adobe
nchhaya@adobe.com

Abstract

Distributed representations of words have shown to be useful to improve the effectiveness of IR systems in many sub-tasks like query expansion, retrieval and ranking. Algorithms like word2vec, GloVe and others are also key factors in many improvements in different NLP tasks. One common issue with such embedding models is that words like *happy* and *sad* appear in similar contexts and hence are wrongly clustered close in the embedding space. In this paper we leverage Aff2Vec, a set of word embeddings models which include affect information, in order to better capture the affect aspect in news text to achieve better results in information retrieval tasks, also such embeddings are less hit by the synonym/antonym issue. We evaluate their effectiveness on two IR related tasks (query expansion and ranking) over the New York Times dataset (TREC-core '17) comparing them against other word embeddings based models and classic ranking models.

1 Introduction

Distributed representations of words, also known as word embeddings, have played a key role in various downstream NLP tasks. Such vector representations place vectors of semantically similar words close in the embedding space, allowing for efficient and effective estimation of word similarity. Word2vec [MCCD13] and

GloVe [PSM14] are among the most widely adopted word embedding models because of their effectiveness in capturing word semantics. One of the advantage of using word embeddings in information retrieval is that they are more effective in capturing query intent and document topics than other local vector representations traditionally used in IR (like TF-IDF vectors). Text tokens in IR don't always overlap with exact words; tokens often coincide with subwords (e.g. generated by stemmers), ngrams, shingles, etc. Therefore word embeddings are also often referred to as *term* embeddings in the context of IR. Term embeddings can be used to rank queries and documents; in such context a dense vector representation for the query is derived and scored against corresponding dense vector representations for documents in the IR system. Query and document vector representations are generated by aggregating term or word embeddings associated with their respective text terms from the query and document texts. Word embeddings can also be used in the query expansion task. Term embeddings are used in such contexts to find good expansion candidates from a global vocabulary of terms (by comparing word vectors), such enriched queries are used to retrieve the documents. Most of recent good performing word embedding models are generated in an unsupervised manner by learning word representations looking at their surrounding contexts. However one issue with word embeddings is that words with about opposite meanings can have very similar contexts, so that, for example, 'happy' and 'sad' may lie closer than they should be in the embedding space, see related efforts in [CLC⁺15] and [NWV16]. In order to mitigate this semantic understanding issue, we propose to use affect-enriched word embedding models (also known as Aff2Vec[KCC18]) for IR tasks, as they outperform baseline word embedding models on word-similarity task and sentiment analysis. Our contribution is the usage of Aff2Vec models as term embeddings for information retrieval in the news domain. Beyond the

Copyright © 2019 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.

In: A. Aker, D. Albakour, A. Barrón-Cedeño, S. Dori-Hacohen, M. Martinez, J. Stray, S. Tippmann (eds.): Proceedings of the NewsIR'19 Workshop at SIGIR, Paris, France, 25-July-2019, published at <http://ceur-ws.org>

synonym-antonym issue we except Aff2Vec models to work well for news IR because of their capability of better capturing writers’ affective attitude towards articles’ text (see section 1.1). We present experiments against standard IR datasets, empirically establishing the utility of the proposed approach.

1.1 Affect scores in news datasets

In order to assess the potential applicability of Aff2Vec embeddings in the context of information retrieval, we run preliminary evaluation of the amount of formality, politeness and frustration contained in common text collections used in information retrieval experiments. For this purpose we leverage the *affect scoring* algorithm that is used for building Aff2Vec embeddings. We extract mean affect scores for formality, politeness and frustration on each dataset. Such an evaluation involves two collection of news: the datasets from TREC core 2018 track, Washington Post articles, and TREC core 2017 track, New York Times articles. Also we extract affect scores from the ClueWeb09 dataset [CHYZ09], containing text of HTML pages crawled from the Web, and the CACM dataset, a collection of titles and abstracts from the CACM journal. Results are reported in table 1.

Dataset	Dataset affect scoring		
	formality	politeness	frustration
NYT	0.7087	0.6291	0.6248
WP	0.7788	0.7456	0.6510
CACM	0.3619	0.1229	0.3511
ClueWeb09	0.4319	0.2708	0.6216

Table 1: Mean affect scores on some common IR datasets

The scores for formality, politeness and frustration extracted on the Ney Work Times and Washington Post articles are generally higher than the ones extracted for CACM and ClueWeb09 datasets, except for the frustration score reported for ClueWeb which is very close to the frustration score extracted for NYT articles. These results suggest that Aff2Vec embeddings should work well on the news domain as they are built to appropriately capture such affective aspects of information.

2 Related work

Dict2vec[TGH17b] builds word embeddings using on-line dictionaries and optimizing an objective function where each word embedding is built via positive sampling of strongly correlated words and negative sampling of weak correlated ones [TGH17a]. In [ZC17], embeddings are optimized using different objective functions in a supervised manner based on lists of

queries and related relevant and non-relevant results. In [FFJ⁺16], word vectors in combination with bilingual dictionaries are used to extract synonyms so that they can be used to expand queries. Documents are represented as bags of vectors generated as mixture of distributions in [RPMG16]. Efforts like [CLC⁺15] and [NWV16] are related to our work in the fact that they can be incorporated in usage of term embeddings in IR tasks. For our ranking scenario, [RGMJ16] is relevant as documents and queries are represented by mixtures of Gaussians over word embeddings, each of the Gaussians centered around centroid learned via e.g. a k-means algorithm. The likelihood of a query with respect to a document is measured by the distance of the query vector from each centroid that document belongs to, using *centroid similarity* or average inter-similarity.

2.1 Aff2Vec: Affect-enriched embeddings [KCC18]

Word representations historically have only captured semantic or contextual information, but ignored other subtle word relationships such as difference in sentiment. Affect refers to the feeling of an emotion or a feeling [Pic97]. Words such as ‘glad’, ‘awesome’, ‘happy’, ‘disgust’ or ‘sad’ can be referred to as affective words. Aff2Vec introduces a post-training approach that introduces ‘emotion’-sensitivity or affect information in word embeddings. Aff2Vec leverages existing affect lexicon such as Warriner’s lexicon [WKB13] which has a list of over 14,000 English words tagged with valence (V), arousal (A), and dominance (D) scores. The affect-enriched embeddings introduced by Aff2Vec are either built on top of vanilla word embeddings i.e. word2vec, GloVe, or paragram or introduced along with counterfitting [MOT⁺16] or retrofitting [FDJ⁺15]. In this work, we leverage these enriched vector spaces too in order to evaluate their performance for standard IR tasks, namely - query expansion and ranking.

3 Word embeddings for query expansion

We leverage word embeddings to perform query expansion in a way similar to [RPMG16]. For each query term q contained in the query text Q , the word embedding model is used to fetch w_q nearest neighbour w_e in the embedding space, so that $\cos(w_e, w_q) > t$, where t is the minimum allowed cosine similarity between two embeddings to consider the word e associated to the vector w_e a good expansion for the word q associated with the query term vector w_q . Upon successful retrieval of an expansion of at least a term q in a query, a new “alternative” query A where q is

substituted by e is created. Consequently the query to be executed on the IR system becomes a *boolean* query of the form $Q \text{ OR } A$. If more than one query term has a valid expansion fetched from the embedding model, all possible combinations of query terms and relative expansion terms is generated. For example, given a query "recent research about AI", if term embeddings output that $\text{nearest}(\text{recent}) = \text{latest}$ with $\text{cos}(\text{recent}, \text{latest}) = 0.8$ bigger than the threshold 0.75, the output query will be composed by two *optional* clauses: "recent research about AI" OR "latest research about AI".

4 Word embeddings for ranking

In order to use word embedding models for ranking we chose to use the averaging word embeddings approach (also known as *AWE*). Each document and query vector is calculated by averaging the word vectors related to each word in documents and query texts. The query / document score is measured by the cosine similarity between their respective averaged vectors, as in other research works like [MNCC16, RGMJ16, RMLH17, GSS17]. In our experiments we used each word TF-IDF vector to normalize (divide) the averaged word embedding for query and document vectors. We observed that using this technique to smooth the sum of the word vectors instead of just dividing it by the number of its words (mean) resulted in better ranking results. This seems in line with the findings from [SLMJ15] which indicate that cosine similarity may be polluted by term frequencies when comparing word embeddings.

5 Experiments

We compare the usage of Aff2Vec word embeddings in the ranking and query expansion task against both *vanilla* embedding models (like word2vec and GloVe) and enriched models like Dict2vec models [TGH17a]. We also present experiments with variants in Aff2Vec: counterfitted and retrofitted models with enriched affect information. All the models used in our experiments are pretrained. To setup our evaluations we use two open source toolkits *Anserini* [YFL17] and *Lucene4IR* [AMH⁺17], both based on Apache Lucene [BMII12]. We run ranking and query expansion experiments on the New York Times articles from the TREC Core '17 track [AHK⁺17] since it's a relevant dataset for the news domain. For the sake of generalizability, we also conduct the same evaluations over the CACM dataset [Fox83], a "classic" dataset for IR. For the case of query expansion we include evaluation using WordNet [Mil95] in order to provide an expansion baseline not based on word embeddings.

5.1 Results

Table 2 shows performance for ranking experiments on the NYT dataset using different embeddings. We observe that usage of term embeddings doesn't give benefits in many cases, classic BM25 and query likelihood retrieval models provide better NDCG than almost all the models except the affect enriched ones. A GloVe retrofitted affect enriched embedding model is the top performing one for NDCG measure. On the other hand none of the term embedding ranking could outperform BM25 on the mean average precision measure.

Ranking experiments on NYT		
Model	NDCG	MAP
BM25	0.4334	0.1977
QL	0.4325	0.1913
NON ENRICHED MODELS		
GloVe	0.4292	0.1883
GloVe.42B.300d	0.4003	0.1690
GloVe.6B.100d	0.4291	0.1911
GloVe.6B.200d	0.4314	0.1964
GloVe.6B.300d	0.4316	0.1946
GloVe.6B.50d	0.4078	0.1760
GloVe-Twitter-100	0.4212	0.1849
GloVe-Twitter-200	0.4242	0.1873
GloVe-Twitter-50	0.4128	0.1798
GloVe-Twitter-25	0.3541	0.1377
w2v-GoogleNews-300	0.4294	0.1922
dict2vec-dim100	0.4101	0.1885
dict2vec-dim200	0.4155	0.1891
dict2vec-dim300	0.4151	0.1899
ENRICHED MODELS		
counterfit-GloVe	0.3980	0.1720
GloVe-retrofitted	0.4216	0.1861
paragram-counterfit	0.3840	0.1580
paragram-74627	0.4337	0.1937
paragram-retrofitted	0.3969	0.1703
paragram-retrofitted-74627	0.3963	0.1698
w2v-76427	0.4328	0.1969
w2v-counterfit-header	0.3972	0.1721
w2v-retrofitted	0.4341	0.1914
AFFECT ENRICHED MODELS		
counterfit-GloVe-affect	0.4311	0.1753
GloVe-affect	0.4594	0.1926
GloVe-retrofitted-affect-555	0.4693	0.1948
paragram-affect	0.4619	0.1969
paragram-counterfit-affect	0.4339	0.1788
w2v-affect	0.4592	0.1926
w2v-counterfit-affect	0.4309	0.1766
w2v-retrofitted-affect	0.4601	0.1911

Table 2: Ranking experiments on TREC Core '17

Table 3 shows performance for query expansion experiments on the NYT dataset using different embed-

dings. We observe that classic BM25 and query likelihood retrieval models provide better NDCG than almost all the models except some of the affect enriched ones. This is in line with what we observed for the ranking task on the same dataset. A GloVe retrofitted affect enriched embedding model is the top performing one for both NDCG and MAP.

Query expansion experiments on NYT		
Model	MAP	NDCG
BM25	0.1977	0.4334
QL	0.1913	0.4325
NON ENRICHED MODELS		
GloVe	0.1951	0.4337
GloVe.42B.300d	0.1947	0.4308
GloVe.6B.100d	0.1903	0.4291
GloVe.6B.200d	0.1947	0.4308
GloVe.6B.300d	0.1947	0.4308
GloVe.6B.50d	0.1799	0.4119
GloVe-Twitter-100	0.1863	0.4218
GloVe-Twitter-200	0.1863	0.4218
GloVe-Twitter-25	0.1391	0.3488
GloVe-Twitter-50	0.1812	0.4147
w2v-GoogleNews-300	0.1947	0.4308
dict2vec-dim100	0.1995	0.4335
dict2vec-dim200	0.1959	0.4315
dict2vec-dim300	0.1957	0.4315
WordNet	0.1977	0.4334
ENRICHED MODELS		
counterfit-GloVe	0.1801	0.4027
GloVe-retrofitted	0.1940	0.4264
paragram-counterfit	0.1663	0.3906
paragram-74627	0.2005	0.4365
paragram-retrofitted	0.1798	0.4012
paragram-retrofitted-74627	0.1798	0.4012
w2v-76427	0.1964	0.4318
w2v-counterfit-header	0.1734	0.3991
w2v-retrofitted	0.1967	0.4368
AFFECT ENRICHED MODELS		
GloVe-affect	0.1947	0.4308
counterfit-GloVe-affect	0.1810	0.4044
GloVe-retrofitted-affect-555	0.2021	0.4421
paragram-affect	0.1977	0.4309
paragram-counterfit-affect	0.1844	0.4094
w2v-affect	0.1940	0.4305
w2v-counterfit-affect	0.1762	0.4029
w2v-retrofitted-affect	0.1971	0.4345

Table 3: Query expansion experiments on TREC Core '17.

Table 4 shows performance for Ranking experiments on the CACM dataset using different embeddings. We observe that usage of term embeddings generally causes steadily higher NDCG and MAP. In particular the paragram embeddings models report

the best results, with affect enriched paragram embeddings reporting both best NDCG and MAP, 0.02 better than non affect enriched paragram embeddings results in both NDCG and MAP.

Ranking experiments on CACM		
Model	NDCG	MAP
BM25	0.3805	0.1947
QL	0.3621	0.2056
NON ENRICHED MODELS		
GloVe.42B.300d	0.3638	0.2007
GloVe.6B.100d	0.4440	0.2722
GloVe.6B.200d	0.4452	0.2732
GloVe.6B.300d	0.4450	0.2730
GloVe.6B.50d	0.4437	0.2720
GloVe-Twitter-100	0.5109	0.3260
GloVe-Twitter-200	0.5138	0.3292
GloVe-Twitter-25	0.5309	0.3217
GloVe-Twitter-50	0.4682	0.2715
w2v-GoogleNews-300	0.3697	0.1960
GloVe	0.4483	0.2760
ENRICHED MODELS		
counterfit-GloVe	0.4563	0.2680
GloVe-retrofitted	0.4507	0.2787
w2v-76427	0.4920	0.3033
w2v-counterfit-header	0.4085	0.2225
w2v-retrofitted	0.3993	0.2350
paragram-counterfit	0.5675	0.3722
paragram-74627	0.5539	0.3541
paragram-retrofitted	0.5263	0.3467
paragram-retrofitted-74627	0.5380	0.3633
AFFECT ENRICHED MODELS		
counterfit-GloVe-affect	0.4247	0.2383
GloVe-affect	0.4326	0.2553
w2v-affect	0.3900	0.2080
w2v-counterfit-affect	0.3791	0.2006
w2v-retrofitted-affect	0.3555	0.1986
paragram-affect	0.5848	0.3986
paragram-counterfit-affect	0.5860	0.3996

Table 4: Ranking experiments on CACM

Table 5 shows performance for query expansion experiments on the CACM dataset using different embeddings. We observe that usage of term embeddings generally causes steadily higher NDCG and MAP. While we expected best results with Aff2Vec models it turned out "vanilla" word2vec model trained on Google News corpus outperformed all the others in NDCG and MAP. On the other hand the best performing enriched model is a retrofitted word2vec model whereas among affect enriched models the GloVe retrofitted one provides the best results.

Query expansion experiments on CACM		
Model	NDCG	MAP
BM25	0.3805	0.1947
QL	0.3621	0.2056
NON ENRICHED MODELS		
WordNet	0.4014	0.2146
GloVe.42B.300d	0.4657	0.2701
GloVe.6B.100d	0.4646	0.2635
GloVe.6B.200d	0.4633	0.2631
GloVe.6B.300d	0.4724	0.2707
GloVe.6B.50d	0.4575	0.2588
GloVe-Twitter-100	0.4500	0.2576
GloVe-Twitter-200	0.4454	0.2524
GloVe-Twitter-25	0.4215	0.2373
GloVe-Twitter-50	0.4422	0.2528
w2v-GoogleNews-300	0.4824	0.2828
GloVe	0.4635	0.2685
ENRICHED MODELS		
counterfit-GloVe	0.4622	0.2661
GloVe-retrofitted	0.4676	0.2723
w2v-76427	0.4366	0.2518
w2v-counterfit-header	0.4557	0.2629
w2v-retrofitted	0.4738	0.2816
paragram-counterfit	0.4661	0.2716
paragram-74627	0.4626	0.2712
paragram-retrofitted	0.4470	0.2636
paragram-retrofitted-74627	0.4486	0.2646
AFFECT ENRICHED MODELS		
counterfit-GloVe-affect	0.4622	0.2673
GloVe-affect	0.4694	0.2734
GloVe-retrofitted-affect-555	0.4722	0.2799
w2v-affect	0.4609	0.2643
w2v-counterfit-affect	0.4579	0.2674
w2v-retrofitted-affect	0.4667	0.2744
paragram-affect	0.4426	0.2586
paragram-counterfit-affect	0.4634	0.2723

Table 5: Query expansion experiments on CACM

6 Conclusions

We present extensive experiments to evaluate the impact of affect-enriched word embeddings for information retrieval over a news corpus, namely ranking and query expansion implemented using open-source toolkits. We show that using affect-enriched models shows a significant improvement for ranking against baseline/vanilla embeddings (20%) as well as other enriched embeddings (2-10%). In case of query expansion, improvement is observed for the NYT dataset but vanilla GloVe embeddings report highest values for the CACM dataset. We believe the semantic structure and vocabulary distribution of the CACM dataset results in this behavior. We plan to extend this work first towards understanding the role of semantic informa-

tion in expansion tasks and then towards building fusion approaches leveraging enriched word vectors with standard IR baselines.

References

- [AHK⁺17] James Allan, Donna Harman, Evangelos Kanoulas, Dan Li, Christophe Van Gysel, and Ellen Vorhees. Trec 2017 common core track overview. In *Proc. TREC*, 2017.
- [AMH⁺17] Leif Azzopardi, Yashar Moshfeghi, Martin Halvey, Rami S Alkhalaf, Krisztian Balog, Emanuele Di Buccio, Diego Ceccarelli, Juan M Fernández-Luna, Charlie Hull, Jake Mannix, et al. Lucene4ir: Developing information retrieval evaluation resources using lucene. In *ACM SIGIR Forum*, volume 50, pages 58–75. ACM, 2017.
- [BMII12] Andrzej Bialecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*, page 17, 2012.
- [CHYZ09] Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. Clueweb09 data set, 2009.
- [CLC⁺15] Zhigang Chen, Wei Lin, Qian Chen, Xiaoping Chen, Si Wei, Hui Jiang, and Xiaodan Zhu. Revisiting word embedding for contrasting meaning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 106–115, 2015.
- [FDJ⁺15] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, 2015.
- [FFJ⁺16] Linnea Fornander, Marc Friberg, Vida Johansson, V Lindh-Hård, Pontus Ohlsson, and Ida Palm. Generating synonyms using word vectors and an easy-to-read corpus. 2016.
- [Fox83] Edward A Fox. Characterization of two new experimental collections in computer and information science containing textual

- and bibliographic concepts. Technical report, Cornell University, 1983.
- [GSS17] Lukas Galke, Ahmed Saleh, and Ansgar Scherp. Word embeddings for practical information retrieval. In *47. Jahrestagung der Gesellschaft für Informatik, Informatik 2017, Chemnitz, Germany, September 25-29, 2017*, pages 2155–2167, 2017.
- [KCC18] Sopan Khosla, Niyati Chhaya, and Kushal Chawla. Aff2vec: Affect-enriched distributional word representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2204–2218, 2018.
- [MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [MNCC16] Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137*, 2016.
- [MOT⁺16] Nikola Mrkšić, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL-HLT*, pages 142–148, 2016.
- [NWX16] Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. *arXiv preprint arXiv:1605.07766*, 2016.
- [Pic97] Rosalind W. Picard. *Affective Computing*. MIT Press, Cambridge, MA, USA, 1997.
- [PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [RGMJ16] Dwaipayan Roy, Debasis Ganguly, Mandar Mitra, and Gareth JF Jones. Representing documents and queries as sets of word embedded vectors for information retrieval. *arXiv preprint arXiv:1606.07869*, 2016.
- [RMLH17] Navid Rekasaz, Bhaskar Mitra, Mi-hai Lupu, and Allan Hanbury. Toward incorporation of relevant documents in word2vec. *arXiv preprint arXiv:1707.06598*, 2017.
- [RPMG16] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*, 2016.
- [SLMJ15] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, 2015.
- [TGH17a] Julien Tissier, Christophe Gravier, and Amaury Habrard. Dict2vec: Learning word embeddings using lexical dictionaries. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 254–263, 2017.
- [TGH17b] Julien Tissier, Christopher Gravier, and Amaury Habrard. Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2017.
- [WKB13] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45(4):1191–1207, 2013.
- [YFL17] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256. ACM, 2017.
- [ZC17] Hamed Zamani and W Bruce Croft. Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 505–514. ACM, 2017.