MOLEMAN: Mention-Only Linking of Entities with a Mention Annotation Network

Nicholas FitzGerald, Jan A. Botha, Daniel Gillick, Daniel M. Bikel,
Tom Kwiatkowski, Andrew McCallum
Google Research
{nfitz,jabot,dgillick,dbikel,tomkwiat,mccallum}@google.com

Abstract

We present an instance-based nearest neighbor approach to entity linking. In contrast to most prior entity retrieval systems which represent each entity with a single vector, we build a contextualized mention-encoder that learns to place similar *mentions* of the same entity closer in vector space than mentions of different entities. This approach allows all mentions of an entity to serve as "class prototypes" as inference involves retrieving from the full set of labeled entity mentions in the training set and applying the nearest mention neighbor's entity label. Our model is trained on a large multilingual corpus of mention pairs derived from Wikipedia hyperlinks, and performs nearest neighbor inference on an index of 700 million mentions. It is simpler to train, gives more interpretable predictions, and outperforms all other systems on two multilingual entity linking benchmarks.

1 Introduction

A contemporary approach to entity linking represents each entity with a textual description d_e, encodes these descriptions and contextualized mentions of entities, m, into a shared vector space using dual-encoders $f(m)$ and $g(d_e)$, and scores each mention-entity pair as the inner-product between their encodings (Botha et al., 2020; Wu et al., 2019). By restricting the interaction between e and m to an inner-product, this approach permits the pre-computation of all $g(d_e)$ and fast retrieval of top scoring entities using maximum inner-product search (MIPS).

Here we begin with the observation that many entities appear in diverse contexts, which may not be easily captured in a single high-level description. For example, Actor Tommy Lee Jones played football in college, but this fact is not captured in the entity description derived from his Wikipedia page (see Figure 1). Furthermore, when new entities need to be added to the index in a zero-shot setting, it may be difficult to obtain a high quality description. We propose that both problems can be solved by allowing the entity mentions themselves to serve as exemplars. In addition, retrieving from the set of mentions can result in more interpretable predictions – since we are directly comparing two mentions – and allows us to leverage massively multilingual training data more easily, without forcing choices about which language(s) to use for the entity descriptions.

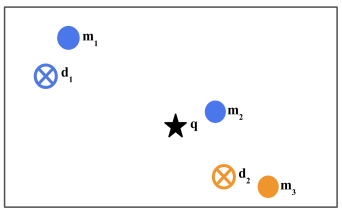
We present a new approach (MOLEMAN[1]) that maintains the dual-encoder architecture, but with the same mention-encoder on both sides. Entity linking is modeled entirely as a mapping between mentions, where inference involves a nearest neighbor search against all known mentions of all entities in the training set. We build MOLEMAN using exactly the same mention-encoder architecture and training data as Model F (Botha et al., 2020). We show that MOLEMAN significantly outperforms Model F on both the Mewsli-9 and Tsai and Roth (2016) datasets, particularly for low-coverage languages, and rarer entities.

We also observe that MOLEMAN achieves high accuracy with just a few mentions for each entity, suggesting that new entities can be added or existing entities can be modified simply by labeling a small number of new mentions. We expect this update mechanism to be significantly more flexible than writing or editing entity descriptions. Finally, we compare the massively multilingual MOLEMAN model to a much more expensive English-only dual-encoder architecture (Wu et al., 2019) on the well-studied TACKBP-2010 dataset (Ji et al., 2010) and show that MOLEMAN is competitive even in this setting.

[1]Mention Only Linking of Entities with a Mention Annotation Network

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 278–285

Query Mention (q): "Harvard, with only five players of the 13, placed captain Vic Gatto and guard {Tom Jones} on the offensive team."



[Q170587]	$\mathbf{d}_1$	"Tommy Lee Jones is an American actor and film director."
	$\mathbf{m}_1$	"{Jones} moved to New York to become an actor in 1969 …"
	$\mathbf{m}_2$	"En Harvard {Jones} jugó como guard ofensivo en el equipo de fútbol americano universitario de 1968."
[Q18152778]	$\mathbf{d}_2$	"Thomas Lee Jones was an American and Canadian football player …"
	$\mathbf{m}_3$	"The Cleveland Browns selected {Tom Jones} …"

Figure 1: Illustration of hypothetical contextualized mention (m) and multilingual description (d) embeddings for the entities 'Tommy Lee Jones (Q170587)' and 'Tom Jones (Q18152778). The query mention [★] pertains to the former's college football career, which is unlikely to be captured by the high-level entity description. A retrieval against descriptions would get this query incorrect, but with indexed mentions gets it correct. Note that prior dual-encoder models that use a single vector to represent each entity are forced to contort the embedding space to solve this problem.

2 Overview

Task definition We train a model that performs entity linking by ranking a set of entity-linked *indexed mentions-in-context*. Formally, let a mention-in-context $\mathbf{x} = [x_1, ..., x_n]$ be a sequence of n tokens from vocabulary $\mathcal{V}$, which includes designated entity span tokens. An *entity-linked* mention-in-context $m^i = (\mathbf{x}^i, e^i)$ pairs a mention with an entity from a predetermined set of entities $\mathcal{E}$. Let $\mathcal{M}_\mathcal{I} = [m^1, ..., m^k]$ be a set of entity-linked mentions-in-context, and let $\texttt{entity}(\cdot) : \mathcal{M}_\mathcal{I} \rightarrow \mathcal{E}$ be a function that returns the entity $e^i \in \mathcal{E}$ associated with m^i, and $\mathbf{x}(\cdot)$ returns the token sequence $\mathbf{x}^i$.

Our goal is to learn a function $\phi(m)$ that maps an arbitrary mention-in-context token sequence m to a fixed vector $\mathbf{h}_m \in \mathcal{R}^d$ with the property that

$$y^* = \texttt{entity}\left(\underset{m' \in \mathcal{M}_I}{\text{argmax}}[\phi(\mathbf{x}(m'))^T \phi(\mathbf{x}_q)]\right)$$

gives a good prediction y^* of the true entity label of a query mention-in-context $\mathbf{x}_q$.

3 Method

3.1 Model

Recent state-of-the-art entity linking systems employ a dual encoder architecture, embedding mentions-in-context and entity representations in the same space. We also employ a dual encoder architecture but we score mentions-in-context (hereafter, mentions) against other mentions, with no consolidated entity representations. The dual encoder maps a pair of mentions (m, m') to a score:

$$s(m, m') = \frac{\phi(m)^T \phi(m')}{\|\phi(m)\| \|\phi(m')\|}$$

where ϕ is a learned neural network that encodes the input mention as a d-dimensional vector.

As in (Févry et al., 2020) and (Botha et al., 2020), our mention encoder is a 4-layer BERT-based Transformer network (Vaswani et al., 2017; Devlin et al., 2019) with output dimension $d = 300$.

3.2 Training Process

3.2.1 Mention Pairs Dataset

We build a dataset of mention pairs using the 104-language collection of Wikipedia mentions as constructed by Botha et al. (2020). This dataset maps Wikipedia hyperlinks to WikiData (Vrandečić and Krötzsch, 2014), a language-agnostic knowledge base. We create mention pairs from the set of all mentions that link to a given entity.

We use the same division of Wikipedia pages into train and test splits used by Botha et al. (2020) for compatibility to the TR2016 test set (Tsai and Roth, 2016). We take up to the first 100k mention pairs from a randomly ordered list of all pairs regardless of language, yielding 557M and 31M training and evaluation pairs, respectively. Of these, 69.7% of pairs involve two mentions from different languages. Our index set contains 651M mentions, covering 11.6M entities.

3.2.2 Hard Negative Mining and Positive Resampling

Previous work using a dual encoder trained with in-batch sampled softmax has improved performance with subsequent training rounds using an auxiliary cross-entropy loss against hard negatives sampled from the current model (Gillick et al., 2019; Wu et al., 2019; Botha et al., 2020). We investigate the effect of such negative mining for MOLEMAN, controlling the ratio of positives to negatives on a per-entity basis. This is achieved by limiting each entity to appear as a negative example at most 10

times as often as it does in positive examples, as done by Botha et al. (2020).

In addition, since MOLEMAN is intended to retrieve the *most similar* indexed mention of the correct entity, we experiment with using this retrieval step to resample the positive pairs used to construct our mention-pair dataset for the in-batch sampled softmax, pairing each mention m with the highest-scoring other mention m' of the same entity in the index set. This is similar to the index refreshing that is employed in other retrieval-based methods trained with in-batch softmax (Guu et al., 2020; Lewis et al., 2020a).

3.2.3 Input Representations

Following prior work (Wu et al., 2019; Botha et al., 2020), our mention representation consists of the page title and a window around the mention, with special mention boundary tokens marking the mention span. We use a total context size of 64 tokens.

Though our focus is on entity mentions, the entity descriptions can still be a useful additional source of data, and allow for zero-shot entity linking (when no mentions of an entity exist in our training set). We therefore experiment with adding the available entity descriptions as additional "pseudo-mentions". These are constructed in a similar way to the mention representations, except without mention boundaries. Organic and psuedo-mentions are fed into BERT using distinct sets of token type identifiers. We supplement our training set with additional mention pairs formed from each entity's description and a random mention, adding 38M training pairs, and add these descriptions to the index, expanding the entity set to 20M.

3.3 Inference

For inference, we perform a distributed brute-force maximum inner product search over the index of training mentions. During this search, we can either return only the top-scoring mention for each entity, which improves entity-based recall, or else all mentions, which allows us to experiment with k-Nearest Neighbors inference (see Section 4.1).

4 Experiments

4.1 Mewsli-9

Table 1 shows our results on the Mewsli-9 dataset compared to the models described by Botha et al. (2020). Model F is a dual encoder which scores entity mentions against entity descriptions, while Model F$^+$ adds two additional rounds of training with hard negative mining and an auxiliary cross-lingual objective. Despite using an identically-sized transformer, and trained on the same data, MOLEMAN outperforms Model F$^+$ when training only on mention pairs, and sees minimal improvement from a further round of training with hard negative and resampled positives (as described in Section 3.2.2). This suggests that training MOLEMAN is a simpler learning problem compared to previous models which must capture all an entity's diverse contexts with a single description embedding. Additionally, we examine a further benefit of indexing multiple mentions per entity: the ability to do top-K inference, and find that top-1 accuracy improves by half a point with k=5.

	I	HN	R@1	R@10	R@100
Model F	D	N	63.0	91.7	97.4
Model F$^+$	D	Y	89.4	96.4	98.2
MGENRE	–	–	90.6	–	–
MOLEMAN	M	N	89.5	97.4	98.3
	B	N	89.6	98.0	99.2
	B	Y	89.9	98.1	99.2
+ k=5	B	Y	90.4	–	–

Table 1: Results on Mewsli-9 compared to the models described by (Botha et al., 2020) and (De Cao et al., 2021). Column I indicates what is being indexed (Descriptions, Mentions, Both), and the HN indicates if additional rounds of Hard Negative training are applied.

We also compare to the recent MGENRE system of De Cao et al. (2021), which performs entity linking using constrained generation of entity names. It should be noted that this work uses an expanded training set that results in fewer zero- and few-shot entities (see De Cao et al. (2021) Table 3).

4.1.1 Per-Language Results

Table 2 shows per-language results for Mewsli-9. A key motivation of Botha et al. (2020) was to learn a massively multilingual entity linking system, with a shared context encoder and entity representations between 104 languages in the Wikipedia corpus. MOLEMAN takes a step further: the indexed mentions from all languages are included in the retrieval index, and can contribute to the prediction in any language. In fact, we find that for 21.4% of mentions in the Mewsli-9 corpus, MOLEMAN's top prediction came from a different language.

Language	R@1	R@10	R@100
ar	+1.1	+0.9	+0.3
de	-0.1	+1.5	+0.5
en	+0.3	+2.8	+2.3
es	-0.2	+1.1	+0.4
fa	+1.1	+0.9	+0.9
ja	+0.8	+1.2	+0.5
sr	-0.1	+0.8	+0.5
ta	+3.7	+1.3	+0.6
micro-avg	+0.2	+1.6	+1.0
macro-avg	+0.8	+1.3	+0.7

Table 2: MOLEMAN results on the Mewsli-9 dataset by language, listed as a delta against Model F^+ (Botha et al., 2020).

4.1.2 Frequency Breakdown

Table 3 shows a breakdown in performance by entity frequency bucket, defined as the number of times an entity was mentioned in the Wikipedia training set. When indexing only mentions, MOLEMAN can never predict the entities in the 0 bucket, but it shows significant improvement in the other frequency bands, particularly in the "few shot" bucket of [1,10). This suggests when introducing new entities to the index, labelling a small number of mentions may be more beneficial than producing a single description. To further confirm this intuition, we retrained MOLEMAN with a modified training set which had all entities in the [1, 10) band of Mewsli-9 removed, and only added to the index at inference time. This model achieved +0.2 R@1 and +5.6 R@10 relative to Model F^+ (which was trained with these entities in the train set). When entity descriptions are added to the index, MOLEMAN outperforms Model F^+ across frequency bands.

4.1.3 Inference Efficiency

Due to the large size of the mention index, nearest neighbor inference is performed using distributed maximum inner-product search. We also experiment with approximate search using ScaNN (Guo et al., 2020). Table 4 shows throughput and recall statistics for brute force search as well as two approximate search approaches that run on a single multi-threaded CPU, showing that inference over such a large index can be made extremely efficient with minimal loss in recall.

4.2 Tsai Roth 2016 Hard

In order to compare against previous multilingual entity linking models, we report results on the "hard" subset of Tsai and Roth (2016)'s cross-lingual dataset which links 12 languages to English Wikipedia. Table 5 shows our results on the same 4 languages reported by Botha et al. (2020). MOLEMAN outperforms all previous systems.

4.3 TACKBP 2010

Recent work on entity linking have employed dual-encoders primarily as a retrieval step before reranking with a more expensive cross-encoder (Wu et al., 2019; Agarwal and Bikel, 2020). Table 6 shows results on the extensively studied TACKBP 2010 dataset (Ji et al., 2010). Wu et al. (2019) used a 24-layer BERT-based dual-encoder which scores the 5.9 million entity descriptions from English Wikipedia, followed by a 24-layer cross-encoder reranker. MOLEMAN does not achieve the same level of top-1 accuracy as their full model, as it lacks the expensive cross-encoder reranking step, but despite using a single, much smaller Transformer and indexing the larger set of entities from multilingual Wikipedia, it outperforms this prior work in retrieval recall at 100.

We also report the accuracy of a MOLEMAN model trained only with English training data, and using an Enlish-only index for inference. This experiment shows that although the multilingual index contributes to MOLEMAN's overall performance, the pairwise training data is sufficient for high performance in a monolingual setting.

5 Discussion and Future Work

We have recast the entity linking problem as an application of a more generic mention encoding task. This approach is related to methods which perform clustering on test mentions in order to improve inference (Le and Titov, 2018; Angell et al., 2020), and can also be viewed as a form of cross-document coreference resolution (Rao et al., 2010; Shrimpton et al., 2015; Barhom et al., 2019). We also take inspiration from recent instance-based language modelling approaches (Khandelwal et al., 2020; Lewis et al., 2020b).

Our experiments demonstrate that taking an instance-based approach to entity-linking leads to better retrieval performance, particularly on rare entities, for which adding a small number of mentions leads to superior performance than a single description. For future work, we would like to explore the application of this instance-based approach to entity knowledge related tasks (Seo et al., 2018; Petroni et al., 2020), and to entity discovery (Ji et al., 2017).

	MOLEMAN (mentions only)		MOLEMAN (+ descriptions)		mGENRE
Freq. bin	R@1	R@10	R@1	R@10	R@1
[0, 1)	-8.3†	-33.9†	-0.2	+18.3	+13.8
[1, 10)	+0.4	+5.6	+1.7	+9.3	-10.4
[10, 100)	+1.9	+3.8	+1.7	+3.7	-3.1
[100, 1k)	+0.1	+1.8	-0.0	+1.9	+0.3
[1k, 10k)	-1.1	+0.7	-1.2	+0.7	+0.6
[10k,+)	+0.7	+0.6	+0.7	+0.5	+2.2
macro-avg	-1.1	-3.6	+0.5	+5.7	+0.6

Table 3: Results from MOLEMAN (with and without the inclusion of entity descriptions) on the Mewsli-9 dataset, by entity frequency in the training set plotted as a delta against Model F^+. †Note that when using mentions only, MOLEMAN scores zero on entities that do not appear in the training set.

	QPS	Latency (ms)	R@1	R@100
Brute-force	9.5	5727	89.9	99.2
ScaNN	8000	2.9	89.9	99.1

Table 4: Max throughput (queries per second), latency (ms per query) and recall for brute force inference and approximate MIPS inference using the ScaNN library (Guo et al., 2020). See Appendix A.3 for further details.

	MF+	MM
de	0.62	0.64
es	0.58	0.59
fr	0.54	0.58
it	0.56	0.59
Avg	0.57	0.60

Table 5: Accuracy results on the TR2016^{hard} test set for Model F^+ (MF+) and MOLEMAN (MM)

Method	**R@1**	**R@100**
AT-Prior	–	89.5
AT-Ext	–	91.7
BM25	–	68.9
Gillick et al. (2019)	–	96.3
Wu et al. (2019)	91.5†	98.3*
MOLEMAN (EN-only)	85.8	98.4
MOLEMAN	87.9	99.1

Table 6: Retrieval comparison on TACKBP-2010. The alias table and BM25 baselines are taken from Gillick et al. (2019). For comparison to Wu et al. (2019), we report R@1 for their "full Wiki, w/o finetune" cross-encoder. Their R@100 model is a dual-encoder fine-tuned on the TACKBP-2010 training set. MOLEMAN is not finetuned.

Acknowledgements

The authors would like to thank Ming-Wei Chang, Livio Baldini-Soares and the anonymous reviewers for their helpful feedback. We also thank Dave Dopson for his extensive help with profiling the brute-force and approximate search inference.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI 2016*.

Oshin Agarwal and Daniel M Bikel. 2020. Entity linking via dual and cross-attention encoders. *arXiv preprint arXiv:2004.03555*.

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2020. Clustering-based inference for zero-shot biomedical entity linking. *arXiv preprint arXiv:2010.11253*.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *ACL 2019*.

Jan A Botha, Zifei Shan, and Dan Gillick. 2020. Entity linking in 100 languages. In *EMNLP 2020*.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021. Multilingual autoregressive entity linking. *arXiv preprint arXiv:2103.12528*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL 2019*.

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. 2020. Empirical evaluation of pretraining strategies for supervised entity linking. In *AKBC 2020*.

Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *CoNLL 2019*.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *ICML 2020*.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *TAC 2010*.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, Cash Costello, and Sydney Informatics Hub. 2017. Overview of tac-kbp2017 13 languages entity discovery and linking. In *TAC 2017*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *ICLR 2020*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *ACL 2018*.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. In *NeurIPS 2020*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS 2020*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2020. KILT: a Benchmark for Knowledge Intensive Language Tasks.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *COLING 2010: Posters*.

Minjoon Seo, Tom Kwiatkowski, Ankur P Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. In *EMNLP 2018*.

Luke Shrimpton, Victor Lavrenko, and Miles Osborne. 2015. Sampling techniques for streaming cross document coreference resolution. In *NAACL 2015*.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *ACL 2016*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS 2017*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP 2020*.

A Appendices

A.1 Training setup and hyperparameters

To isolate the impact of representing entities with multiple mention embeddings, we follow the training methodology and hyperparameter choices presented in Botha et al. (2020) (Appendix A).

We train MOLEMAN using in-batch sampled softmax (Gillick et al., 2018) using a batch size of 8192 for 500k steps, which takes about a day. Our model is implemented in Tensorflow (Abadi et al., 2016), using the Adam optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2017) with the mention encoder preinitialized from a multilingual BERT checkpoint[2]. All model training was carried out on a Google TPU v3 architecture[3].

[2] github.com/google-research/bert/multi_cased_L-12_H-768_A-12

[3] cloud.google.com/tpu/docs/tpus

A.2 Datasets Links

- Mewsli-9: http://goo.gle/mewsli-dataset
- TR2016^{h}ard: cogcomp.seas.upenn.edu/page/resource_view/102
- TACKBP-2010: https://catalog.ldc.upenn.edu/LDC2018T16

A.3 Profiling Details

The brute-force numbers we've reported are the theoretical maximum throughput for computing 300D dot-products on an AVX-512 processor running at 2.2Ghz, and are thus an overly optimistic baseline. Practical implementations, such as the one in ScaNN, must also compute the top-k and rarely exceed 70% to 80% of this theoretical limit. The brute-force latency figure is the minimum time to stream the database from RAM using 144 GiB/s of memory-bandwidth. In practice, we ran distributed brute-force inference on a large cluster of CPUs, which took about 5 hours.

The numbers for ScaNN are empirical single-machine benchmarks of an internal solution that uses the open-source ScaNN library [4] on a single 24-core CPU. We use ScaNN to search a multi-level tree that has the following shape: $78,000 => 83:1 => 105:1$ (687.3 million datapoints). We used a combination of several different anisotropic vector quantizations that combine 3, 6, 12, or 24 dimensions per 4-bit code, as well as re-scoring with an `int8`-quantization.

A.4 Expanded experimental results

Tables 7 and 8 present complete numerical comparisons between MOLEMAN and Model F^{+} on Mewsli-9.

[4] https://github.com/google-research/google-research/tree/master/scann

	Model F+			MOLEMAN (mentions only)			MOLEMAN (+ descriptions)		
Language	R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@10	R@100
ar	92.3	97.7	99.1	93.4	98.6	99.0	93.4	98.6	99.4
de	91.5	97.3	99.0	91.3	98.2	98.9	91.5	98.9	99.5
en	87.2	94.2	96.7	87.4	95.9	97.4	87.4	97.0	99.3
es	89.0	97.4	98.9	88.7	98.1	98.8	88.7	98.5	99.3
fa	91.8	97.4	98.7	93.5	98.5	99.1	92.9	98.3	99.6
ja	87.8	95.6	97.6	88.7	96.2	97.0	88.5	96.8	98.0
sr	92.6	98.2	99.2	92.2	98.7	99.5	92.5	99.0	99.7
ta	87.6	97.4	98.9	91.5	98.4	99.1	91.3	98.6	99.5
micro-avg	89.4	96.4	98.2	89.5	97.4	98.3	89.6	98.0	99.2
macro-avg	89.8	96.9	98.5	90.6	97.8	98.5	90.6	98.2	99.3

Table 7: Results on the Mewsli-9 dataset by language.

		Model F+			MOLEMAN (mentions only)			MOLEMAN (+description)		
Bin	Queries	R@1	R@10	R@100	R@1	R@10	R@100	R@1	R@10	R@100
[0, 1)	3,198	8.3	33.9	62.7	0.0	0.0	0.0	8.1	52.2	74.7
[1, 10)	6,564	57.7	80.8	91.3	58.1	86.4	93.3	59.4	90.1	96.5
[10, 100)	32,371	80.4	92.8	96.7	82.2	96.5	98.8	82.1	96.5	98.9
[100, 1k)	66,232	89.6	96.6	98.2	89.7	98.4	99.5	89.6	98.5	99.5
[1k, 10k)	78,519	92.9	98.4	99.3	91.9	99.2	99.8	91.8	99.1	99.8
[10k, +)	102,203	94.1	98.8	99.4	94.8	99.4	99.6	94.8	99.3	99.5
micro-avg		89.4	96.4	98.2	89.5	97.4	98.3	89.6	98.0	99.2
macro-avg		70.5	83.5	91.3	69.4	80.0	81.8	70.9	89.3	94.8

Table 8: Results on the Mewsli-9 dataset, by entity frequency in the test set.