Robust Candidate Generation for Entity Linking on Short Social Media Texts

Liam Hebert
University of Waterloo
l2hebert@uwaterloo.ca

Raheleh Makki
Twitter
rmakki@twitter.com

Shubhanshu Mishra
Twitter
smishra@twitter.com

Hamidreza Saghir
Twitter
hsaghir@twitter.com

Anusha Kamath
Twitter
akamath@twitter.com

Yuval Merhav
Twitter
ymerhav@twitter.com

Abstract

Entity Linking (EL) is the gateway into Knowledge Bases. Recent advances in EL utilize dense retrieval approaches for Candidate Generation, which addresses some of the shortcomings of the Lookup based approach of matching NER mentions against pre-computed dictionaries. In this work, we show that in the domain of Tweets, such methods suffer as users often include informal spelling, limited context, and lack of specificity, among other issues. We investigate these challenges on a large and recent Tweets benchmark for EL, empirically evaluate lookup and dense retrieval approaches, and demonstrate a hybrid solution using long contextual representation from Wikipedia is necessary to achieve considerable gains over previous work, achieving 0.93 recall.

1 Introduction

Entity Linking (EL) is the task of linking mentions to their corresponding entities in a Knowledge Base (KB) such as Wikidata. EL is commonly formulated in three sequential steps: Named Entity Recognition (NER), where mentions are identified, Candidate Generation, where a list of possible entity candidates is generated, and Entity Disambiguation, where a final candidate is selected.

Earlier EL works relied on alias tables (dictionary from strings to possible Wikidata entities; often associated with a score) and key-word based retrieval methods (Spitkovsky and Chang, 2012; Logeswaran et al., 2019; Pershina et al., 2015). However, these approaches suffer on noisy text, such as short-form Tweets. An example of a difficult Tweet would be "***Liam*** *is a gr8 ML Researcher*" where the desired span to link would be "Liam". Here, an alias-based approach would only retrieve entities based on the span "Liam", of which there are 8,350 different Wikidata entities containing that name. Without the context of "gr8 ML Researcher", it quickly becomes unfeasible to find the correct candidate. Furthermore, alias based approaches are also heavily dependent on the spans retrieved, where the retrieved span must be exactly present in the alias table in order to be found (Spitkovsky and Chang, 2012; Logeswaran et al., 2019; Pershina et al., 2015). This presents a challenge due to the difficulties of NER systems on noisy social media text (Lample et al., 2016; Mishra et al., 2020).

More recently, BERT-based dense entity retrieval approaches have shown to produce SOTA results on news datasets such as TACKBP-2010 and Mewsli-9 (Wu et al., 2020; FitzGerald et al., 2021; Botha et al., 2020). Dense retrieval approaches rely on relevant context around the mention, which is abundant in long and clean documents such as news, but often absent or brief in noisy and short user-generated text, such as that found on Twitter.

Prior works that focus on social media linking, such as Tweeki (Harandizadeh and Singh, 2020), used small, annotated datasets and did not study the more recent dense retrieval approaches.

Recently, Twitter researchers released an end-to-end entity linking benchmark for Tweets called TweetNERD. It is the largest and most temporally diverse open-sourced dataset benchmark on Tweets (Mishra et al., 2022). Excited by the availability of this benchmark, we study the application of recent linking methods on this large and noisy user generated data. We empirically evaluate sparse and dense retrieval approaches on this data and describe the challenges and design choices of building a robust linking system for Tweets.

Our main contributions are as follows: **(A)** To the best of our knowledge, we are the first study to compare dense retrieval, sparse retrieval, and lookup based approaches for Entity Linking in a social media setting, which makes our work relevant for the research community interested in processing noisy user generated text. **(B)** We assess the robustness of dense retrieval techniques in the presence of span detection errors coming from NER

Proceedings of the 2022 COLING Workshop: The 8th Workshop on Noisy User-generated Text (W-NUT 2022), pages 83–89

systems for social media text. This is a common problem for social media datasets as the top NER F1 score for social media datasets is significantly lower than other domains (Strauss et al., 2016). **(C)** We assess the impact of using short Wikidata entity descriptions against the longer Wikipedia descriptions for representing candidates, and highlight the significant loss in performance from using shorter descriptions for social media text. This is relevant as many recent dense retrieval methods for generic Entity Linking have proposed using short descriptions from Wikidata for candidate representations. **(D)** Our analysis is the first to explore sparse and dense retrieval on the largest and most temporally diverse Entity Linking dataset for Tweets called TweetNERD (Mishra et al., 2022). **(E)** Finally, through quantitative and qualitative analysis, we assert the complimentary nature of candidates generated by lookup and dense retrieval based approaches. This asserts the validity of our hybrid approach towards candidate generation and is reflected in significant performance improvement by using hybrid candidate generation for Entity Linking.

2 Methodology

2.1 Knowledge Base

To represent our KB, we followed prior work and retrieved a July 2022 download of English Wikipedia[1] (Wu et al., 2020; De Cao et al., 2020). However, Wikipedia also includes miscellaneous pages or pages that refer to multiple entities, such as disambiguation pages and "list of" pages. An example of such a page is "List of Birds of Canada" [2], which describes 696 distinct birds, each with their own respective Wikipedia page. To detect these pages, we retrieve the "instance of" category of each entity from Wikidata, which classifies each Wikipedia entity into distinct categories. Using this information, we reduce the entity set from 56.8M to 6.5M Wikidata entities.

2.2 Span Detection

We observe the performance of our systems utilizing the Gold Spans provided by TweetNERD (Table 1) and compare that to using NER-based spans that reflect a more realistic use-case. The NER model is trained on Tweets from TweetNERD and is similar to the models described in Lample et al. (2016); Mishra et al. (2020).

2.3 Candidate Generation

2.3.1 Dense Retrieval

Our dense retrieval approach retrieves candidates based on the similarity of tweet and entity embeddings. This is done by utilizing two separate language models to encode the semantic content of Tweets and Entities respectively. Our approach is motivated by Wu et al. (2020), which utilized a similar strategy on a clean news corpus. Given a Tweet t with mention span s and entity e^i, we create dense embeddings as

$$T^s = BERT_T([CLS]\ t_l^s\ [M_1]\ span^s\ [M_2]\ t_r^s) \quad (1)$$

$$E^i = BERT_E([CLS]\ title^i\ [M_3]\ desc^i) \quad (2)$$

where $BERT_T$ and $BERT_E$ are two separate language models, t_l^s and t_r^s refer to the text to the left and right of the desired mention span s, and $title^i$ and $desc^i$ are the Wikipedia title and first ten sentences of the respective entity page. Finally, $[M_1]$, $[M_2]$, $[M_3]$ are special tokens to denote the separation of each of the fields in the input.

Given these dense embeddings, we rank the pairing of entities e to Tweet t by computing the dot product between their corresponding CLS representations. During inference, we pre-compute the embeddings for every entity in our knowledge base and index them using fast k nearest neighbour search provided by FAISS (Johnson et al., 2021). We refer to this approach as Dense.

2.3.2 Sparse Retrieval

We utilize a traditional lookup-based approach for finding candidates as used by many prior works (Harandizadeh and Singh, 2020). Specifically, we map surface forms to Wikipedia page candidates from the English Wikipedia parse of DBPedia Spotlight and rank candidates given $p(entity|surfaceForm)$. We also include Wikidata aliases and labels as both have been found previously to be beneficial for identifying named entities (Mishra and Diesner, 2016; Singh et al., 2012; Mendes et al., 2011) and entity candidates in text (Mendes et al., 2011; Mishra et al., 2022; Singh et al., 2012). We refer to this approach as Lookup.

[1] This was the latest version at the time of writing

[2] https://en.wikipedia.org/wiki/List_of_birds_of_Canada

Table 1: Candidate Generation using Gold Spans (R@16)

Data Split	Dense	Lookup	BM25	Hybrid
Academic	0.783	0.741	0.221	**0.916**
OOD	0.772	0.847	0.556	**0.933**
Overall	0.779	0.717	0.362	**0.930**

3 Results

3.1 Experimental Setup

We use TweetNERD for training and evaluation. It consists of 340K+ Tweets linked to entities in Wikidata (Mishra et al., 2022). We follow the authors' setup and evaluate on TweetNERD-Academic and TweetNERD-OOD (out of domain), while the rest of the data is used for training. For Dense retrieval we use pre-trained BLINK[3] encoders which are trained on Wikipedia text and FAISS (Johnson et al., 2021) for indexing candidate embeddings. We compare that to a Lookup based system (Section 2.3.2) and a BM25 baseline (Yang et al., 2018). For BM25, we utilize Wikipedia abstracts as candidate documents and mention spans as queries.

In all experiments, we limit our retrieved candidates set for Dense and BM25 to the top 16 entities due to observed diminishing returns (Figure 1). For Lookup, we retrieve all exact match candidates since they are not explicitly ranked. As a result, the performance of Lookup reflects an upper-bound of the performance of that method. The average number of retrieved Lookup candidates is 19 while the median of 4, reflecting the long tail distribution of retrieved candidates per span.

3.2 Candidate Generation

We begin by evaluating the impact of dense retrieval on Candidate Generation. Since we constrain our dense retrieval methods to 16 candidates, we measure Recall @16 of our various systems.

3.2.1 Gold Spans

We first observe the performance of our systems utilizing the Gold Spans provided by TweetNERD (Table 1). Contrasting Lookup and Dense, we can see that Dense outperforms on the Academic split by 4 points whereas Lookup outperforms on the Out-of-Domain split by 7.5 points. In addition, we see that our trivial BM25 baseline falls significantly behind with 0.221 recall on the Academic set and 0.556 on the OOD set.

[3]https://github.com/facebookresearch/BLINK

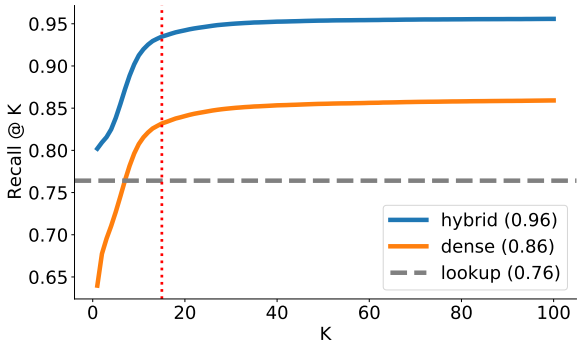

Figure 1: Recall @ K of Dense, Lookup and Hybrid using Gold Spans

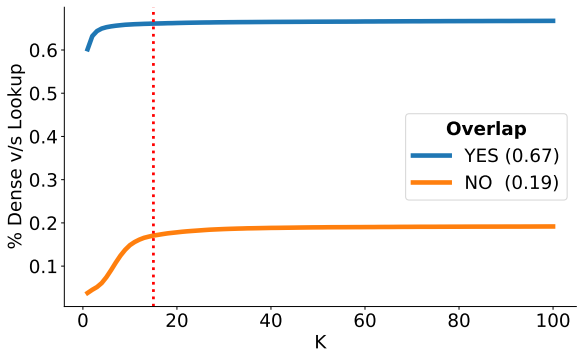

Figure 2: Overlap and Distinction of Dense v/s Lookup using Gold Spans

Upon further investigation, we find that Dense and Lookup methods produce mutually exclusive results. On the Academic dataset, we find that Dense retrieved 7719 unique correct candidates whereas lookup retrieved 5268 unique correct candidates (Table 2 and Table 3). Leveraging these differences and inspired by van Hulst et al. (2020), we take the union of both methods as a Hybrid approach. This approach yielded a significant **+17.5 recall** increase over Lookup and **+13.3 recall** increase over Dense on the Academic split. In Figure 1, we show the change in Recall for all approaches as K increases. We can see that the benefit of retrieving more Dense candidates plateaus after 16

Table 2: Unique Correct Candidates using Gold Spans

Data Split	Dense	Lookup	BM25
Academic	**7,719**	5,268	1,043
OOD	1,055	**2,664**	1,495
Overall	**8,774**	7,932	2,538

Table 3: Candidate Overlap Across Lookup, Dense and BM25 using Gold Spans

Lookup	Dense	BM25	counts	prop
Y	Y	Y	16,310	0.30
Y	Y	N	19,810	0.36
Y	N	Y	2,190	0.04
Y	N	N	3,566	0.06
N	Y	Y	1,079	0.02
N	Y	N	8,298	0.15
N	N	Y	361	0.01
N	N	N	3,386	0.06

Table 4: Candidate Generation using NER Spans (R@16)

Data Split	Dense	Lookup	BM25	Hybrid
Academic	0.761	0.613	0.164	**0.880**
OOD	0.754	0.757	0.440	**0.903**
Overall	0.759	0.715	0.245	**0.887**

candidates. However, we also find that candidates retrieved by Lookup and Dense continue to be mutually exclusive despite the larger candidate set (Figure 2). This illustrates that the performance plateau is not due to overlap in candidate sets but rather that both methods produce vastly different candidates. We investigate these differences in Section 3.2.3.

3.2.2 NER Spans

Next, to reflect a real-life use-case, we investigate performance of our system on NER spans. Here, we annotate each Tweet using the NER service described in Section 2.2. We capture the recall performance of our systems by evaluating the set of all retrieved candidates against the set of gold entities (Table 4). Here, we can see the benefits of Dense retrieval where Dense achieved similar performance on NER spans as utilizing gold spans. This is contrasted by Lookup, which realized a significant drop in performance. This is likely due to inaccuracies in our NER system, which can return spans that do not have exact entries in our pre-computed table.

We also see a continuing trend of complementary results between Dense retrieval and Lookup. Here, Dense and Unique retrieved 8362 and 4711 unique correct entities on the Academic set, respectively (Table 5). By combining the retrieved candidates from both sets, we can increase the performance of Lookup by $\approx$ **26.7 points** on all splits.

Table 5: Unique Correct Candidates using NER Spans

Data Split	Dense	Lookup	BM25
Academic	**8,362**	4,711	983
OOD	1,263	**2,448**	1,496
Overall	**9,625**	7,159	2,479

3.2.3 Qualitative Analysis

During our experiments, we found significant differences between the candidates retrieved by Dense retrieval and Lookup retrieval. We find that these differences can largely be categorized into span ambiguity, spelling, and the presence of context.

An example of a TweetNERD Tweet requiring context due to span ambiguity would be *"Wiz and **Amber**, Rihanna and Chris, Beyonce and jay-z #grammyscouples"* where the desired span is the word "Amber".

In our results, we found that Lookup returned many entities containing the name "Amber", such as "AMBER Alert" (Q1202607) and "Amber, Rajasthan, India" (Q8197166), but not the correct entity "Amber Rose" (Q290856). To the reader, it is clear upon reading the entire Tweet that the meaning does not concern a rescue service or city, but rather celebrities who have dated someone named "Wiz". This is contrasted by Dense retrieval, which returned the correct entity, but also similar entities such as celebrity "Amber Benson" (Q456862). Furthermore, we can see in the Wikipedia entity description of Amber Rose that she had been married to Wiz Khalifa, information that would not be present in the lookup table.

However, the presence of context can also be detrimental and misleading when taken literally. An example of such a TweetNERD Tweet would be *"No one here remembers The Marine and the **12 Rounds**."* where the desired span is "12 Rounds".

In this case, Dense retrieval returned incorrect candidates such as "12 Gauge Shotgun" (Q2933934), instead of "12 Rounds" the movie (Q245187). However, this was mitigated by Lookup, which accurately found the correct entity. We hypothesize that the context of "Marines" combined with "12 Rounds" misleads the Dense model to retrieve entities related to weaponry, instead of matching the literal title as Lookup did.

4 Conclusion

In this work, we have evaluated the usage of sparse and dense retrieval techniques towards candidate generation on social media text. In our qualitative and quantitative experimentation, we have highlighted the complementary strengths of both methods. Combined, our hybrid approach achieves significant improvements on TweetNERD, a large temporally diverse dataset for entity linking on Tweets. We also demonstrate the improvements that dense retrieval translates to improved downstream entity linking performance using both gold and NER based spans.

There are also a few directions for future work. First, in this work we focused on the Candidate Generation step for Entity Linking. While we report preliminary results for the Entity Disambiguation step in Appendix Section A, future work could explore efficient ways to disambiguate the candidates retrieved from our hybrid approach. Second, future work could expand our evaluation beyond the English Tweets found in TweetNERD and develop a multi-lingual solution. Third, it is important to note that there are significant linguistic differences between the formal text found on Wikipedia and informal speech on Twitter. Recent work has explored leveraging mentions as entity descriptions, which could be applied to Twitter text to bridge this gap (FitzGerald et al., 2021).

Overall, our work highlights the best practices for improving entity linking on short and noisy social media text. We hope this work inspires future entity linking efforts on this challenging domain.

References

Jan A Botha, Zifei Shan, and Dan Gillick. 2020. Entity linking in 100 languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Nicholas FitzGerald, Dan Bikel, Jan Botha, Daniel Gillick, Tom Kwiatkowski, and Andrew McCallum. 2021. MOLEMAN: Mention-only linking of entities with a mention annotation network. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 278–285, Online. Association for Computational Linguistics.

Bahareh Harandizadeh and Sameer Singh. 2020. Tweeki: Linking named entities on twitter to a knowledge graph. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*.

J. Johnson, M. Douze, and H. Jegou. 2021. Billion-scale similarity search with gpus.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460.

Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. 2011. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*.

Shubhanshu Mishra and Jana Diesner. 2016. Semi-supervised named entity recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 203–212, Osaka, Japan. The COLING 2016 Organizing Committee.

Shubhanshu Mishra, Sijun He, and Luca Belli. 2020. Assessing demographic bias in named entity recognition. In *Proceedings of the AKBC Workshop on Bias in Automatic Knowledge Graph Construction, 2020*. arXiv.

Shubhanshu Mishra, Aman Saini, Raheleh Makki, Sneha Mehta, Aria Haghighi, and Ali Mollahosseini. 2022. TweetNERD - End to End Entity Linking Benchmark for Tweets.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015.

Valentin I. Spitkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3168–3175, Istanbul, Turkey. European Language Resources Association (ELRA).

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine De Marneffe, and Wei Xu. 2016. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144.

Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407.

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using lucene. *J. Data and Information Quality*, 10(4).

A Entity Disambiguation

To evaluate end-to-end EL performance, we conduct preliminary experiments by training a disambiguation model using the candidate set retrieved from our retrieval methods. Once we generate entity candidates, we score each <Mention, Entity> pair for each Tweet using common mention-entity Lookup based features (e.g., mention count per entity), entity only based features (e.g., Wikipedia page rank), and contextual mention-entity features generated by comparing the mention embedding in the text against the candidate entity description embedding. We train our model to identify the correct entity for each span among the retrieved candidates. Our architecture and features are like the ones described in Kolitsas et al. (2018) with the major difference being the usage of a BERT based encoder instead of BiLSTM.

While our focus is candidate generation, reporting end-to-end performance is important since improvement in candidate generation does not necessarily translate to end-to-end improvement. Dense, unlike Lookup, can retrieve the right candidate even when the mention span is missing due to NER errors, however our disambiguation system currently still requires a span in order to link a mention.

Dataset Split	Dense	Lookup
Academic	**0.617**	0.566
OOD	**0.605**	0.568
Overall	**0.610**	0.567

Table 6: F1 of Entity Disambiguation using NER Spans

Description	Recall	Precision	F1
Lookup			
Short	0.484	**0.686**	0.567
Long	0.543	0.628	0.582
Dense			
Short	0.299	0.249	0.272
Long	**0.613**	0.607	**0.610**

Table 7: Ablation Experiments on Entity Disambiguation

Table 6 shows the F1 score of our disambiguation model using candidates retrieved by our proposed methods. Our results demonstrate that the increased recall brought by Dense candidates have translated into increased end-to-end F1 on all splits when compared to Lookup, achieving a 0.04 F1 gain. Furthermore, we can see the largest difference on the Academic split, where Dense achieved 0.051 higher F1 then our lookup-based approach.

B Ablation Study

A core part of our methodology is how we represent entities. In our proposed approach, we utilize Wikipedia descriptions, which provide a verbose but rich description of entities. We refer to these descriptions as "Long" descriptions. To evaluate the impact of these descriptions on Dense and Lookup retrieval, we conduct an ablation study where we evaluate utilizing Wikidata descriptions. These descriptions are much shorter and terse, often never exceeding 5-6 words. An example of such a description would be "species of bird", which is shared by 23 828 different bird entities [4]. We refer to these Wikidata descriptions as "Short" descriptions.

The results of our ablation study can be seen in Table 7. While we see an overall improvement when utilizing Long descriptions, the most significant impact can be seen on dense retrieval, where

[4] https://www.wikidata.org/w/index.php?search=species+of+bird

we see a leap of F1 performance from 0.272 to 0.610. Furthermore, we can also see that Lookup can still perform well when utilizing Short descriptions, achieving our highest precision result.

There are a few reasons for these results. Due to the k-nearest neighbour nature of Dense retrieval, entities that are retrieved by this method are often very semantically similar. This was demonstrated in Section 3.2.3, where Dense retrieval returned a list of actors when trying to link to an actor mention. However, since short descriptions are often shared between related entities ("species of bird"), often the same description would appear in the retrieved list. This is contrasted by Lookup, where the list of retrieved entities is related only by mentioned name. As a result, the entities are typically much more diverse (AMBER Alert vs Amber Rose) and thus easier to disambiguate with shorter descriptions.