Minimalist Entity Disambiguation for Mid-Resource Languages

Benno Kruit
Vrije Universiteit Amsterdam
De Boelelaan 1105, 1081 HV
Amsterdam, Netherlands
b.b.kruit@vu.nl

Abstract

For many languages and applications, even though enough data is available for training Named Entity Disambiguation (NED) systems, few off-the-shelf models are available for use in practice. This is due to both the large size of state-of-the-art models, and to the computational requirements for recreating them from scratch. However, we observe that in practice, acceptable models can be trained and run with far fewer resources. In this work, we introduce `MiniNED`, a framework for creating small NED models from medium-sized datasets. The resulting models can be tuned for application-specific objectives and trade-offs, depending on practitioners' requirements concerning model size, frequency bias, and out-of-domain generalization. We evaluate the framework in nine languages, and achieve reasonable performance using models that are a fraction of the size of recent work.

1 Introduction

Motivation and Problem. Named Entity Disambiguation (NED), is the task of linking pre-identified entity names to their corresponding entries in a knowledge base, such as Wikipedia. As a crucial component of Entity Linking (EL) applications, it has been extensively studied for almost two decades. Presently, powerful state-of-the-art EL systems are available based on (English or multilingual) Neural Language Models (Botha et al., 2020; Wu et al., 2020; van Hulst et al., 2020; De Cao et al., 2022). Unfortunately, for many languages and applications there are few off-the-shelf models that are easy to distribute, customize, or use in constrained practical settings. This is due to the large size of trained models, as well as the computational requirements for creating them from scratch. As a result, EL systems are often unavailable or too large to run for many applications.

Approach. In this work, we focus on *mid-resource* languages (e.g. Persian, Japanese, and Tamil), which have some linguistic resources and tools available but not many (Ortiz Suárez et al., 2020). We claim that for many of these languages, simpler and smaller models can perform well enough with careful trade-off analyses. Our main observation is that here the range of reasonable NED performance is quite narrow. In other words, the lower bound (i.e. simply predicting the most commonly linked entity for a given name) and the non-zero-shot upper bound (i.e. perfectly disambiguating all names that are seen in training) are very close together. Based on this insight, we demonstrate that small NED models can achieve acceptable performance with limited resources. We examine the trade-offs between model size and performance for different configurations and highlight the importance of language-specific phenomena, such as morphological differences, in determining optimal parameter settings. We argue that the tuning of NED models for mid-resource languages requires careful consideration and can only be done sustainably on small models.

Contribution. We introduce and evaluate `MiniNED`[1], a Python library for creating NED models from Wikipedia data in many languages. We show that much simpler models than state-of-the-art systems can achieve acceptable performance in practice. We also show how our framework allows practitioners to control model complexity and adjust for specific use-cases, while maintaining performance.

2 Observations

Examining the Mewsli-9 benchmark (Botha et al., 2020), we can make several observations about the distribution of data that is available in Wikipedia[2] for training NED models.

[1] https://github.com/bennokr/miniNED

[2] All experiments were performed on Wikipedia dumps from 2022-03-01. In Mewsli-9, we replace English by Dutch due to our focus on mid-resource languages.

Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP), pages 299–306

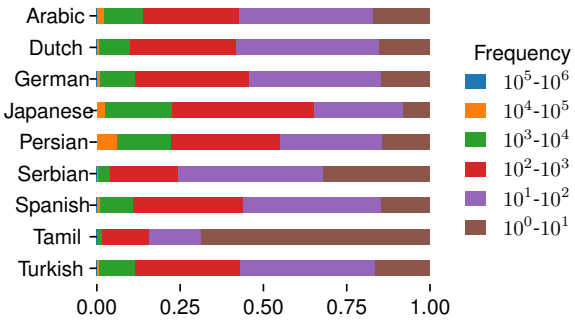


Figure 1: Distribution of entity hyperlink frequencies in Wikipedia, for names from Mewsli-9.

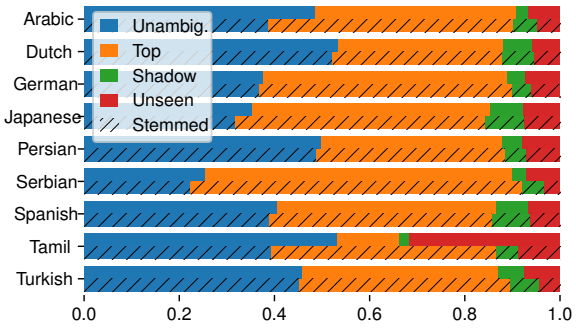


Figure 2: Proportion of benchmark mention instances per ambiguity category, Mewsli-9 dataset. Hatched bars indicates the stemming of names, which decreases unseen cases, but increases ambiguity.

In Figure 1, we show the distribution of how often entities from the benchmark data are hyperlinked on Wikipedia. While we observe a typical long-tailed distribution, most entities can still be observed between 10-1000 times. For these languages, this provides enough training data to learn to disambiguate entity mentions from their context.

In Figure 2, however, we observe that the baseline performance of simply predicting the *top* most commonly linked entity for a given ambiguous name (combined with straightforwardly linking *unambiguous* names) can already achieve relatively high performance. Additionally, many entity-name pairs (which we will refer to as *mentions*) in the benchmark data cannot be observed in training at all; such *unseen* cases would require zero-shot generalization. Consequently, the upper and lower bound for simple models are very close together. Thus, we may comclude that the main challenge lies in predicting *shadowed* entity mentions (Provatorova et al., 2021), which share a surface form with more popular entities. Due to the overwhelming imbalance of training instances for shadowed entities, particular attention should be given to selecting appropriate training data and to the assumptions that underly the model.

Another observation is that language-specific phenomena make a big difference. The distribution of observed ambiguity changes when names are *stemmed*. Stemming removes word inflections, which increases the ambiguity of names. We can see that due to morphological differences, the effect of stemming is very different per language. Overall, stemming decreases the number of unseen mentions, but also widens the range of ambiguous names.

Finally, Wikipedia hyperlink data is noisy (Gerlach et al., 2021), as it includes links to disambiguation pages and incorrect entity links. This becomes a larger problem when less data is available (as for low-resource languages and domains). We argue that optimally tuning the training pipeline to overcome this noise (by fixing this data or patching the model) can only be done sustainably when the model itself is simple.

3 Approach

We train multinomial logistic regression classifiers with hashed Bag-of-Word features, which are trained to rank candidate entities using Vowpal Wabbit (vw, Langford et al., 2007). The candidates are created by filtering entity mention counts from Wikipedia dumps using heuristics. These heuristics identify valid surface forms based on their appearance on disambiguation pages, their string similarity to the entity labels, and the entropy of the prior probability distribution of hyperlink targets. Re-

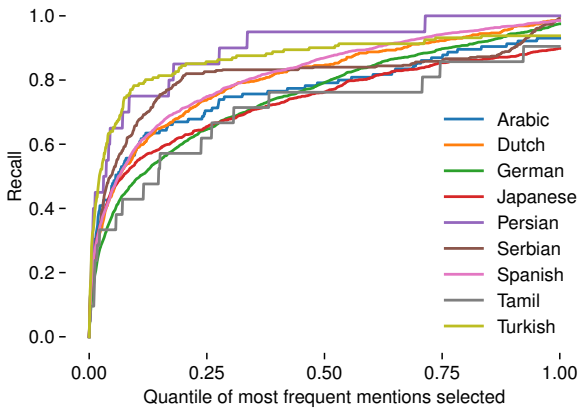


Figure 3: Maximum attainable recall of observed ambiguous mentions given filtering thresholds. When keeping the top 25% most frequent mention-entity instances, the maximum attainable recall on *ambiguous* benchmark instances is 55-85% .

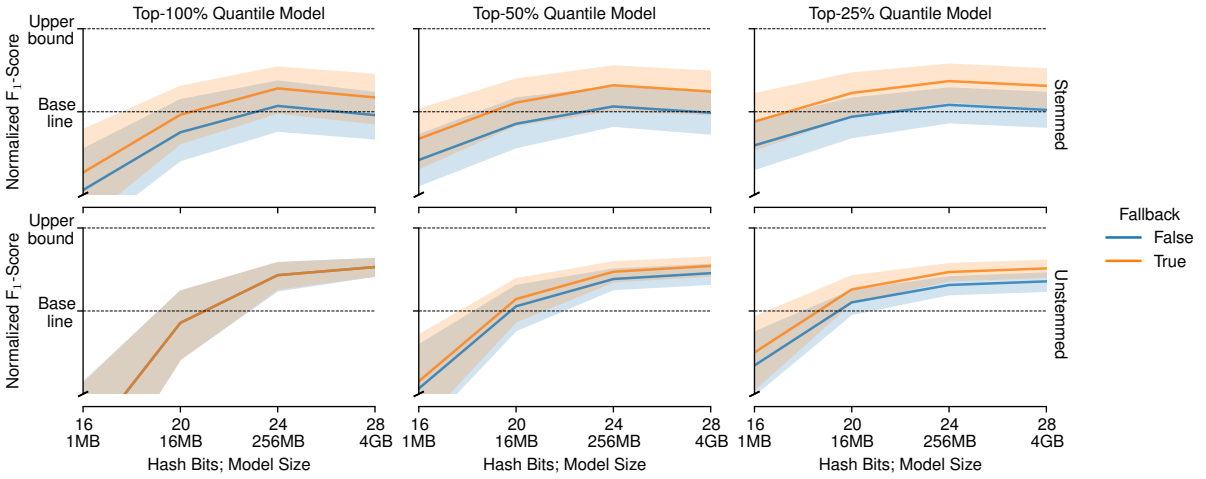


Figure 4: Distribution (mean and 95% confidence interval) of normalized micro F_1-scores. Per-language scores are transformed w.r.t. the baseline (most frequent target of entity labels) and the upper bound on observed mentions (Section 2). *Subplots*: Performance improves logistically with model size, with strongly diminishing returns. *Colors*: Using a fallback to the baseline for unobserved names recovers most lost performance for stemmed and quantile-filtered models. *Rows*: Stemmed models have higher variety in performance between languages, and may overfit when large. *Columns*: Quantile-filtered models require fewer parameters. Raw results in Appendix A.

garding the entropy, a cutoff threshold determines when a long, flat prior distribution is discarded (e.g. specific villages for the anchor text "village"). This also leads to anchor texts with flat distributions of very similarly frequent targets to be discarded entirely (e.g. for strings such as "click here").

The candidate set can be further filtered by only selecting a percentage (i.e. quantile) of most frequently observed ambiguous mentions. In Figure 3, we show the tradeoff between candidate mention filtering and the maximum recall that a model trained to disambiguate between these candidates could achieve. We observe that selecting only the most frequently mentioned entities results in quick gains with only a fraction of the candidate set size. However, we also observe that our pre-processing heuristics discard valid mentions for some languages, so that even on the full set of candidates, perfect recall cannot be attained.

The vw model size is controlled by the number of bits of the feature hash, which also works as regularisation to prevent overfitting. By exchanging single coefficients per mention-feature pair for smaller models with more hash collisions, we are able to find the optimal tradeoff between model size and accuracy using a hyperparameter sweep. Although the features are hashed, the model can still be audited by keeping track of feature hashes for specific analyses. This can be useful for explaining individual predictions (showing which context words have a strong influence), or examining the coefficients that are used to disambiguate a single surface form.

	Baseline	Fallback	Best $\text{Model}_{\text{bits}}$	Upper Bound
Arabic	.87	.87/.88	$.82_{28}/.\mathbf{89}_{28}$	.93/.91
Dutch	.63	.77/.77	$.77_{28}/.\mathbf{78}_{28}$	.84/.83
German	.80	**.85**/.84	$.84_{28}/.\mathbf{85}_{28}$	.90/.88
Japanese	.80	.83/**.84**	$.81_{28}/.83_{28}$	.91/.89
Persian	.85	.86/.86	$.\mathbf{88}_{28}/.\mathbf{88}_{24}$	.91/.90
Serbian	.76	**.84**/.80	$.83_{28}/.80_{28}$	.89/.83
Spanish	.71	.80/.80	$.78_{28}/.\mathbf{81}_{28}$	.89/.88
Tamil	.61	.74/.62	$.\mathbf{75}_{24}/.63_{24}$	.77/.64
Turkish	.80	**.84**/.81	$.80_{28}/.81_{28}$	.91/.87

Table 1: Micro F_1-scores (stemmed / unstemmed). *Baseline*: most frequent target of entity labels. *Fallback*: most frequent target of pre-processed hyperlinks. *Best Model*: score & bits of highest-scoring model configuration. *Upper Bound*: Perfect performance on observed mentions.

4 Evaluation

Our analysis compares models of different sizes and candidate filtering thresholds, and the effect of stemming in different languages. We modify the Mewsli-9 benchmark to discard links to disambiguation pages and list pages (statistics in Appendix B), and we generate the Dutch data using the scripts provided by Botha et al. (2020). We train on lowercased mentions that occur more than once, which are filtered by discarding names which both (1) have less than 10% of tokens appear in an entity label on Wikidata and (2) have a high candidate entropy (> 1 nat), except if they are used as

Figure 5: Inspecting strongest feature coefficients in the Dutch model for the name "Utrecht", which among others may refer to a city, province, town in South Africa, university, or football club.

main links on disambiguation pages. For stemming, we use PersianStemmer (Taghi-Zadeh et al., 2015), MeCab for Japanese (Kudo, 2006), and Snowball for other languages (Porter, 2001).

In Figure 4, we report normalized disambiguation micro-F_1 scores, where per-language scores are transformed with respect to the baseline (the most frequent target of entity labels) and the upper bound (on observed mentions). Unnormalized results are presented per language for best-performing models in Table 1 and in Appendix A.

We observe that the effect of stemming and the trade-off between model size and performance is different per language, but clear trends are visible, with diminishing returns of model sizes above a few hundred MB.

Explainability and Denoising. By keeping track of which features hash to which parameters for a set of example instances, we can visualize which context words have a strong influence on model predictions (Figure 5). This is useful for improving models for which the training data may have been noisy, allowing practitioners to modify preprocessing pipelines or employ data re-labeling efforts.

5 Related Work

Mewsli-9 was introduced by Botha et al. (2020) and used for evaluation by De Cao et al. (2022). Our evaluation results are not directly comparable to theirs because we remove links to disambiguation pages and list pages.

Some EL systems for mid-resource languages exist. Most prominently, DBpedia Spotlight (Daiber et al., 2013) publish EL models for some languages, but these are not tunable for size. Tsai and Roth (2016) perform cross-lingual wikification using multilingual embeddings; we plan to replace our BoW features by such embeddings in future work. Pappu et al. (2017) train lightweight multilingual entity linking models, but not for mid-resource languages. Gerlach et al. (2021) focus on precision, while we focus on F_1-scores and model size.

Modern EL models often combine Mention Detection (MD) and NED end-to-end. Hachey et al. (2013) describe the interplay of MD and NED in English EL. Ling et al. (2015) extend this description, and make similar observations to ours about NED baselines. Kolitsas et al. (2018) are the first to train end-to-end neural EL models, improved later by De Cao et al. (2021). These efforts were extended to multilingual models by Botha et al. (2020) and De Cao et al. (2022). Such end-to-end neural models require many GPU-hours to train, making it impossible to tune them sustainably for specific applications. In contrast, we focus on the smallest possible NED models, because small MD models can be achieved with the use of gazetteers and their interplay may be optimized by tuning.

6 Conclusion and Future Work

We introduce and evaluate `MiniNED`, a Python library for creating NED models from Wikipedia data in many languages. We show that much simpler models than state-of-the-art systems can achieve acceptable performance in practice. We also show how our framework allows practitioners to control model complexity and adjust for specific use-cases, while maintaining performance.

For future research, we expect this approach to also be useful when incorporating more background knowledge about the entities with richer feature representations and using weak supervision (Orr et al., 2021). Also, the tradeoffs that this work analyses are strongly related to the difficulty of determining the appropriate granularity of EL systems (Van Erp and Groth, 2020). For example, the knowledge base that the mentions are linked to may distinguish between municipalities as admin-

istrative entities and the cities within them, while for many applications this distinction is not relevant. In the future we aim to add support for tuning models to the desired levels of granularity.

Acknowledgements

This research is partially funded by Huawei Amsterdam Research Center.

I would like to thank Thiviyan Thanapalasingam, Majid Mohammedi and Erman Acar for their early feedback on language-specific models, and the members of the VU Amsterdam *Knowledge in AI* and *Learning & Reasoning* groups, Winston Wansleeben, Rens Hassfeld and Mara Spadon for their feedback on drafts of this work.

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*, pages 7833–7845.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *International Conference on Semantic Systems*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

Martin Gerlach, M. Miller, Rita Ho, Kosta Harlan, and Djellel Eddine Difallah. 2021. Multilingual entity linking system for wikipedia with a machine-in-the-loop approach. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130–150.

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.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer.

John Langford, Lihong Li, and Alex Strehl. 2007. Vowpal wabbit online learning project.

Xiao Ling, Sameer Singh, and Daniel S. Weld. 2015. Design Challenges for Entity Linking. *Transactions of the Association for Computational Linguistics*, 3(2011):315–328.

Laurel J. Orr, Megan Leszczynski, Neel Guha, Sen Wu, Simran Arora, Xiao Ling, and Christopher Ré. 2021. Bootleg: Chasing the tail with self-supervised named entity disambiguation. In *11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15, 2021*.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Aasish Pappu, Roi Blanco, Yashar Mehdad, Amanda Stent, and Kapil Thadani. 2017. Lightweight multilingual entity extraction and linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM 17, New York, NY, USA. ACM.

Martin F Porter. 2001. Snowball: A language for stemming algorithms.

Vera Provatorova, Samarth Bhargav, Svitlana Vakulenko, and Evangelos Kanoulas. 2021. Robustness evaluation of entity disambiguation using prior probes: the case of entity overshadowing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10501–10510.

Hossein Taghi-Zadeh, Mohammad Hadi Sadreddini, Mohammad Hasan Diyanati, and Amir Hossein Rasekh. 2015. A new hybrid stemming method for persian language. *Digital Scholarship in the Humanities*, 32(1):209–221.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598.

Marieke Van Erp and Paul Groth. 2020. Towards entity spaces. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2129–2137.

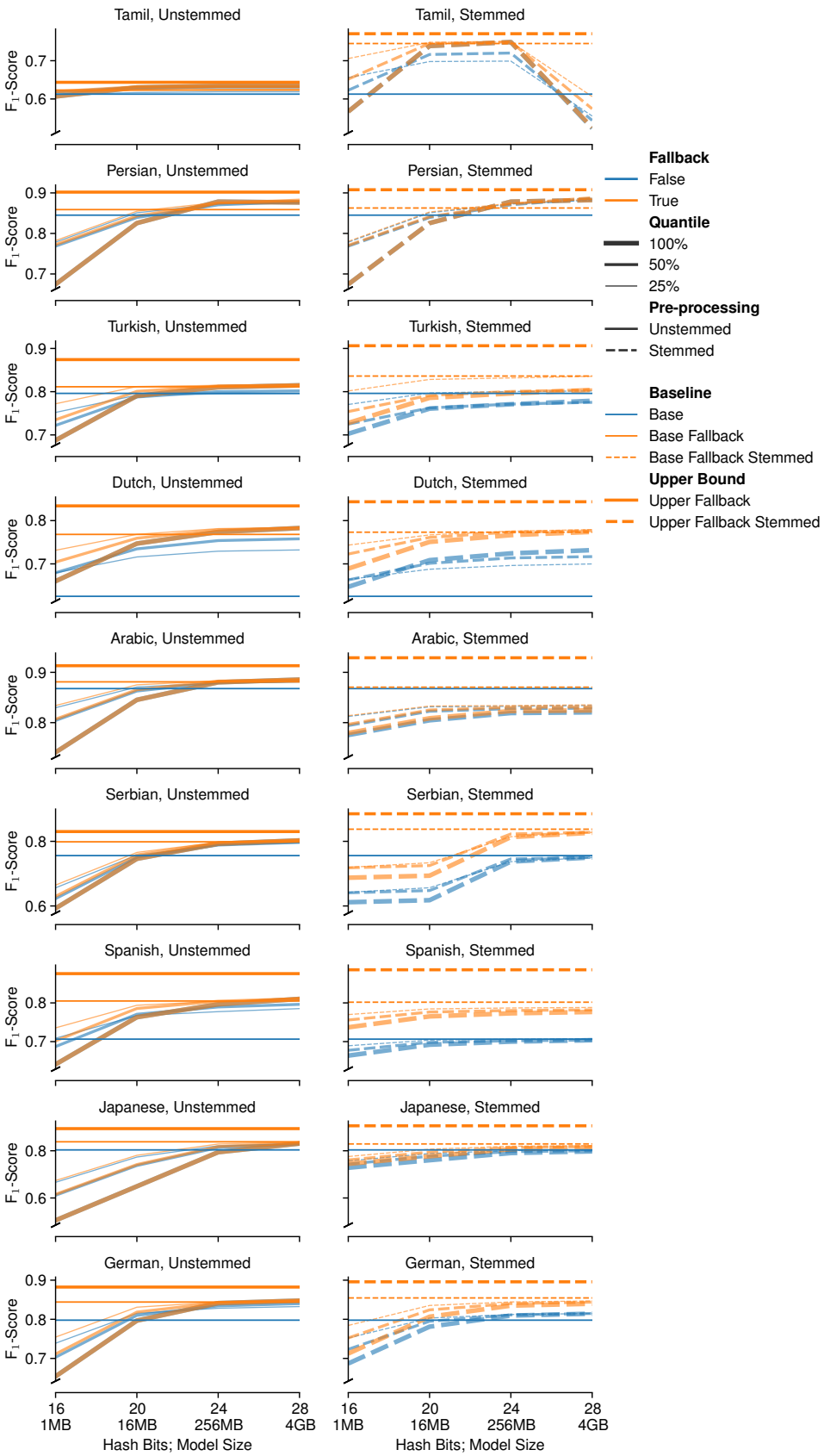
Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. REL: An Entity Linker Standing on the Shoulders of Giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200, Virtual Event China. ACM.

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*, pages 6397–6407.

Appendix

A Evaluation Results

Micro-F_1 scores per model size. *Line width*: Candidate selection filtering quantile. *Color*: Use of fallback to baseline (most frequent target).



B Mewsli-9 Modification

	Disambig	Listpage	Total
Arabic	201	4	5964
Dutch	562	16	11924
German	1907	76	64807
Japanese	605	54	34214
Persian	5	0	515
Serbian	773	7	35536
Spanish	1923	105	55431
Tamil	28	1	2683
Turkish	164	5	5661

Table 2: Statistics on discarded Mewsli-9 links out of the total original dataset