Goodwill Hunting: Analyzing and Repurposing Off-the-Shelf Named Entity Linking Systems

Karan Goel[*]
Stanford University

Laurel Orr
Stanford University

Nazneen Fatema Rajani
Salesforce Research

Jesse Vig
Salesforce Research

Christopher Ré
Stanford University

Abstract

Named entity linking (NEL) or mapping “strings" to “things" in a knowledge base is a fundamental preprocessing step in systems that require knowledge of entities such as information extraction and question answering. In this work, we lay out and investigate two challenges faced by individuals or organizations building NEL systems. Can they directly use an off-the-shelf system? If not, how easily can such a system be repurposed for their use case? First, we conduct a study of off-the-shelf commercial and academic NEL systems. We find that most systems struggle to link rare entities, with commercial solutions lagging their academic counterparts by 10%+. Second, for a use case where the NEL model is used in a sports question-answering (QA) system, we investigate how to close the loop in our analysis by repurposing the best off-the-shelf model (BOOTLEG) to correct sport-related errors. We show how tailoring a simple technique for patching models using weak labeling can provide a 25% absolute improvement in accuracy of sport-related errors.

1 Introduction

Named entity linking (NEL), the task of mapping from “strings” to “things” in a knowledge base, is a fundamental component of commercial systems such as information extraction and question answering (Shen et al., 2015). Given some text, NEL systems perform contextualized linking of text phrases, called *mentions*, to a knowledge base. If a user asks her personal assistant “How long would it take to drive a Lincoln to Lincoln”, the NEL system underlying the assistant should link the first mention of “Lincoln” to the car company, and the second “Lincoln” to Lincoln in Nebraska, in order to answer correctly.

As NEL models have direct impact on the success of downstream products (Peters et al., 2019), all major technology companies deploy large-scale NEL systems; e.g., in Google Search, Apple Siri and Salesforce Einstein. While these companies can afford to build custom NEL systems at scale, we consider how a smaller organization or individual could achieve the same objectives.

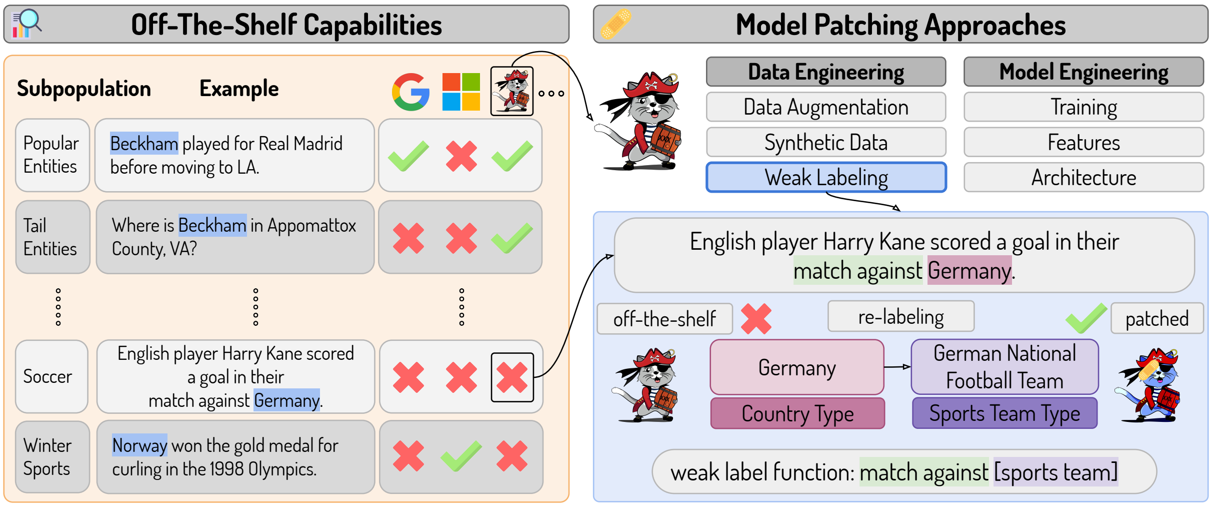
We start with a simple question: how would someone, starting from scratch, build an NEL system for their use case? Can existing NEL systems be used off-the-shelf, and if not, can they be repurposed with minimal engineer effort? Our “protagonist" here must navigate two challenging problems, as shown in Figure 1:

1. **Off-the-shelf capabilities.** Industrial NEL systems provide limited transparency into their performance, and the majority of academic NEL systems are measured on standard benchmarks biased towards popular entities (Steinmetz et al., 2013). However, prior works suggest that NEL systems struggle on so-called “tail" entities that appear infrequently in data (Jin et al., 2014; Orr et al., 2020). As the majority of user queries are over the tail (Bernstein et al., 2012; Gomes, 2017), it is critical to understand the extent to which NEL systems struggle on the tail in off-the-shelf academic and commercial systems.

2. **Repurposing systems.** If off-the-shelf systems are inadequate on the tail or other relevant subpopulations, how difficult is it for our protagonist to develop a customized solution without building a system from scratch? Can they treat an existing NEL model as a black box and still modify its behavior? When faced with designing a NEL system with desired capabilities, prior work has largely focused on developing new systems (Sevgili et al., 2020; Shen et al., 2014; Mudgal et al., 2018). The question of how to guide or “patch” an existing NEL system without changing its architecture, features, or training strategy—what we call *model*

[*] E-mail: kgoel@cs.stanford.edu

Proceedings of NAACL HLT 2021: IndustryTrack Papers, pages 205–213
June 6–11, 2021. ©2021 Association for Computational Linguistics



Off-The-Shelf Capabilities
Subpopulation
Example
Popular Entities
Beckham played for Real Madrid before moving to LA.
Tail Entities
Where is Beckham in Appomattox County, VA?
Soccer
English player Harry Kane scored a goal in their match against Germany.
Winter Sports
Norway won the gold medal for curling in the 1998 Olympics.
Model Patching Approaches
Data Engineering
Data Augmentation
Synthetic Data
Weak Labeling
Model Engineering
Training
Features
Architecture
English player Harry Kane scored a goal in their match against Germany.
off-the-shelf
re-labeling
patched
Germany
Country Type
German National Football Team
Sports Team Type
weak label function: match against [sports team]

Figure 1: Challenges faced by individuals or small organizations in building NEL systems. *(left)* the fine-grained performance of off-the-shelf NEL systems varies widely—struggling on tail entities and sports-relevant subpopulations—making it likely that they must be repurposed for use; *(right)* for a sports QA application where no off-the-shelf system succeeds, the best-performing model (BOOTLEG) can be treated as a black box and successfully patched using weak labeling. In the example, a simple rule re-labels training data to discourage the BOOTLEG model from predicting a country entity (“Germany") when a clear sports-relevant contextual cue (“match against") is present.

engineering—remains unaddressed.

In response to these questions, we investigate the limitations of existing systems and the possibility of repurposing them:

1. **Understanding failure modes (Section 3).** We conduct the first study of open-source academic and commercially available NEL systems. We compare commercial APIs from MICROSOFT, GOOGLE and AMAZON to open-source systems BOOTLEG (Orr et al., 2020), WAT (Piccinno and Ferragina, 2014) and REL (van Hulst et al., 2020) on subpopulations across 2 benchmark datasets of WIKIPEDIA and AIDA (Hoffart et al., 2011). Supporting prior work, we find that most systems struggle to link rare entities, are sensitive to entity capitalization and often ignore contextual cues when making predictions. On WIKIPEDIA, commercial systems lag their academic counterparts by 10%+ recall, while MICROSOFT outperforms other commercial systems by 16%+ recall. On AIDA, a heuristic that relies on entity popularity (POP) outperforms all commercial systems by 1.5 F1. Overall, BOOTLEG is the most consistent system.

2. **Patching models (Section 3.2).** Consider a scenario where our protagonist wants to use a NEL system as part of a downstream QA model answering sport-related queries; e.g., “When did England last win the FIFA world cup?”. All models underperform on sport-relevant subpopulations of AIDA; e.g., BOOTLEG can fail to predict national sports teams despite strong sport-relevant contextual cues, favoring the country entity instead. We therefore take the best system, BOOTLEG, and show how to correct undesired behavior using *data engineering* solutions—model agnostic methods that modify or create training data. Drawing on simple strategies from prior work in weak labeling, which uses user-defined functions to weakly label data (Ratner et al., 2017), we relabel standard WIKIPEDIA training data to patch these errors and finetune the model on this relabeled dataset. With this strategy, we achieve a 25% absolute improvement in accuracy on the mentions where a model predicts a country rather than a sports team.

We believe these principles of understanding fine-grained failure modes in the NEL system and correcting them with data engineering apply to large-scale industrial pipelines where the NEL model or its embeddings are used in numerous downstream products.

2 Named Entity Linking

Given some text, NEL involves two steps: the identification of all entity mentions (*mention ex-*

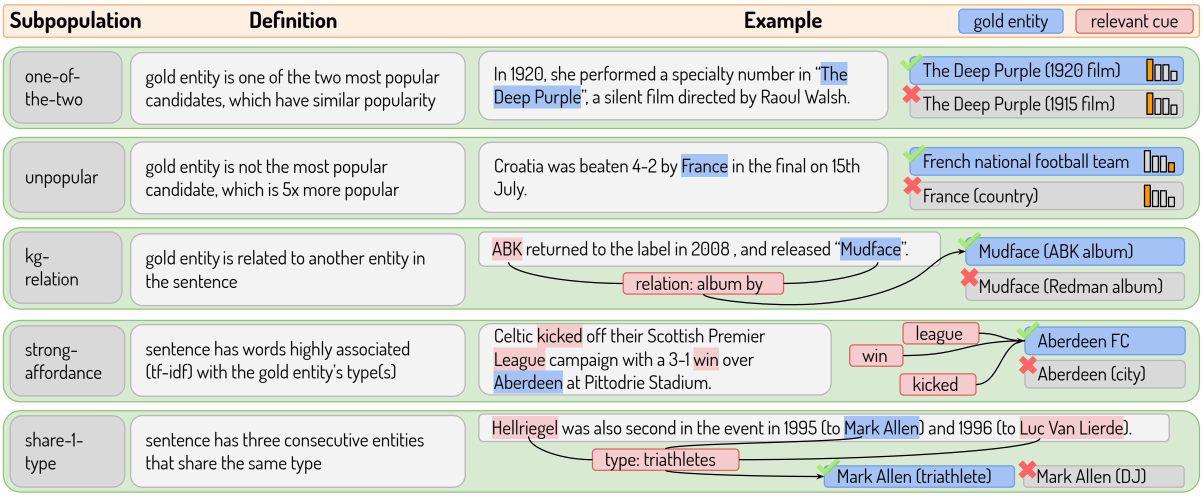

Subpopulation
Definition
Example
gold entity
relevant cue
one-of-the-two
gold entity is one of the two most popular candidates, which have similar popularity
In 1920, she performed a specialty number in "The Deep Purple", a silent film directed by Raoul Walsh.
The Deep Purple (1920 film)
The Deep Purple (1915 film)
unpopular
gold entity is not the most popular candidate, which is 5x more popular
Croatia was beaten 4-2 by France in the final on 15th July.
French national football team
France (country)
kg-relation
gold entity is related to another entity in the sentence
ABK returned to the label in 2008 , and released "Mudface".
relation: album by
Mudface (ABK album)
Mudface (Redman album)
strong-affordance
sentence has words highly associated (tf-idf) with the gold entity's type(s)
Celtic kicked off their Scottish Premier League campaign with a 3-1 win over Aberdeen at Pittodrie Stadium.
league
win
kicked
Aberdeen FC
Aberdeen (city)
share-1-type
sentence has three consecutive entities that share the same type
Hellriegel was also second in the event in 1995 (to Mark Allen) and 1996 (to Luc Van Lierde).
type: triathletes
Mark Allen (triathlete)
Mark Allen (DJ)

Figure 2: Subpopulations analyzed on the WIKIPEDIA dataset, along with their definitions and examples. We consider five subpopulations inspired by Orr et al. (2020).

traction), and contextualized linking of these mentions to their corresponding knowledge base entries (*mention disambiguation*). For example, in "What ingredients are in a Manhattan", the mention "Manhattan" links to `Manhattan (cocktail)`, not `Manhattan (borough)` or `The Manhattan Project`. Internally, most systems have an intermediate step that generates a small set of possible candidates for each mention (*candidate generation*) for the disambiguation model to choose from.

Given the goal of building a NEL system for a specific use case, we need to answer two questions: (1) what are the failure modes of existing systems, and (2) can they be repurposed, or "patched", to achieve desired performance.

3 Understanding Failure Modes

We begin by analyzing the fine-grained performance of off-the-shelf academic and commercial systems for NEL.

Setup. To perform this analysis, we use Robustness Gym (Goel et al., 2021b), an open-source evaluation toolkit for analyzing natural language processing models. We evaluate all NEL systems by considering their performance on *subpopulations*, or subsets of data that satisfy some condition.

Systems. We use 3 commercially available APIs: (i) GOOGLE Cloud Natural Language API (Google) , (ii) MICROSOFT Text Analytics API (Microsoft) , and (iii) AMAZON Comprehend API (Amazon)[1]. We compare to 3 state-of-the-art systems: (i) BOOTLEG, a self-supervised system, (ii) REL, which combines existing state-of-the-art approaches, (iii) WAT an extension of the TAGME (Ferragina and Scaiella, 2010) linker. We also compare to a simple heuristic (iv) POP, which picks the most popular entity among candidates provided by BOOTLEG.

Datasets. We compare methods on examples drawn from two datasets: (i) WIKIPEDIA, which contains $100,000$ entity mentions mined from gold anchor links across $37,492$ sentences from a November 2019 Wikipedia dataset, and (ii) AIDA, the AIDA test-b benchmark dataset[2].

Metrics. As WIKIPEDIA is sparsely labeled (Ghaddar and Langlais, 2017), we compare performance on recall. For AIDA, we use Macro-F1, since AIDA provides a more dense labeling of entities.

Results. Our results for WIKIPEDIA and AIDA are reported in Figures 3, 4 respectively.

3.1 Analysis on WIKIPEDIA

Subpopulations. In line with Orr et al. (2020), we consider 4 groups of examples — *head, torso, tail and toe* — that are based on the popularity of the entities being linked. Intuitively, head examples involve resolving popular entities that occur frequently in WIKIPEDIA, torso examples have medium popularity while tail examples correspond to entities that are seen rarely. Toe entities are a subset of the tail that are almost never seen. We con-

[1] AMAZON performs named entity recognition (NER) to identify entity mentions in text, so we use it in conjunction with a simple string matching heuristic to resolve entity links.

[2] REL uses AIDA for training, so we exclude it.

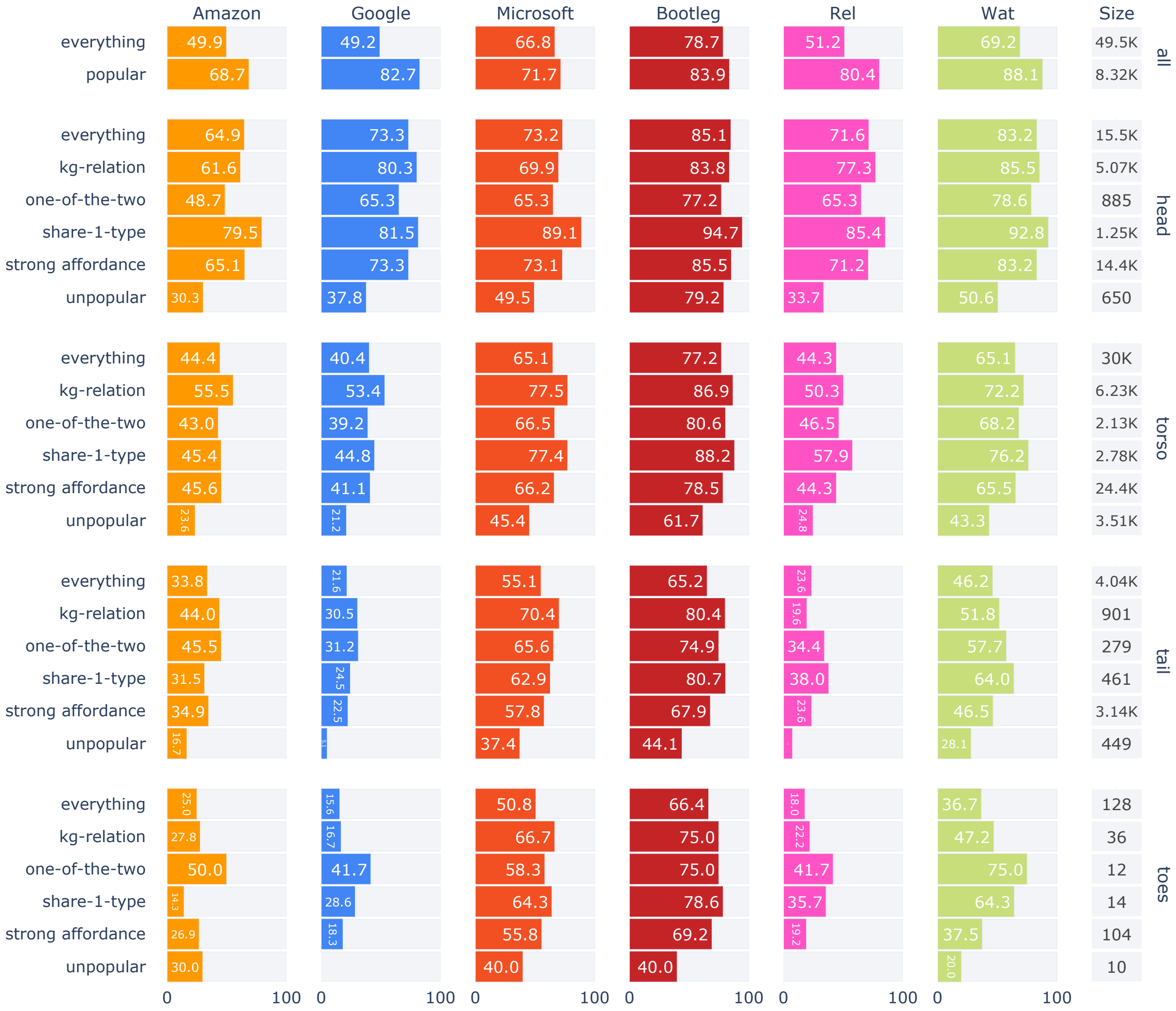


Figure 3: Robustness Report (Goel et al., 2021b) for NEL on Wikipedia, measuring recall.

sider 5 subpopulations inspired by Orr et al. (2020), described in Figure 2 with examples. These subpopulations require close attention to contextual cues such as relations, affordances and types.

We also consider aggregate performance on the entire dataset (**everything**), and **globally popular** entities, which are examples where the entity mention is in the top 800 most popular entity mentions.

BOOTLEG is best overall. Overall, BOOTLEG outperforms other systems by a wide margin, with a 12-point gap to the next best system (MICROSOFT), while MICROSOFT in turn outperforms other commercial systems by more than 16 points.

Performance degrades on rare entities. For all systems, performance on head slices is substantially better than performance on tail/toe slices. BOOTLEG is the most robust across the set of slices that we consider. Among commercial systems, GOOGLE and AMAZON struggle on tail and torso entities e.g. GOOGLE from 73.3 points on *head* to 21.6 points on *tail*, while MICROSOFT's performance degrades more gracefully. GOOGLE is adept at globally popular entities, where it outperforms MICROSOFT by more than 11 points.

3.2 Analysis on AIDA

Subpopulations. We consider subpopulations that vary by: (i) fraction of capitalized entities, (ii) average popularity of mentioned entities, (iii) number of mentioned entities; (iv) sports-related topic.

Overall performance. Similar to WIKIPEDIA, BOOTLEG performs best, beating WAT by 1.3%, with commercial systems lagging by 11%+.

Sensitivity to capitalization. Both GOOGLE and MICROSOFT are sensitive to whether the entity mention is capitalized. GOOGLE's performance goes from 54.1% on sentences where all mentions

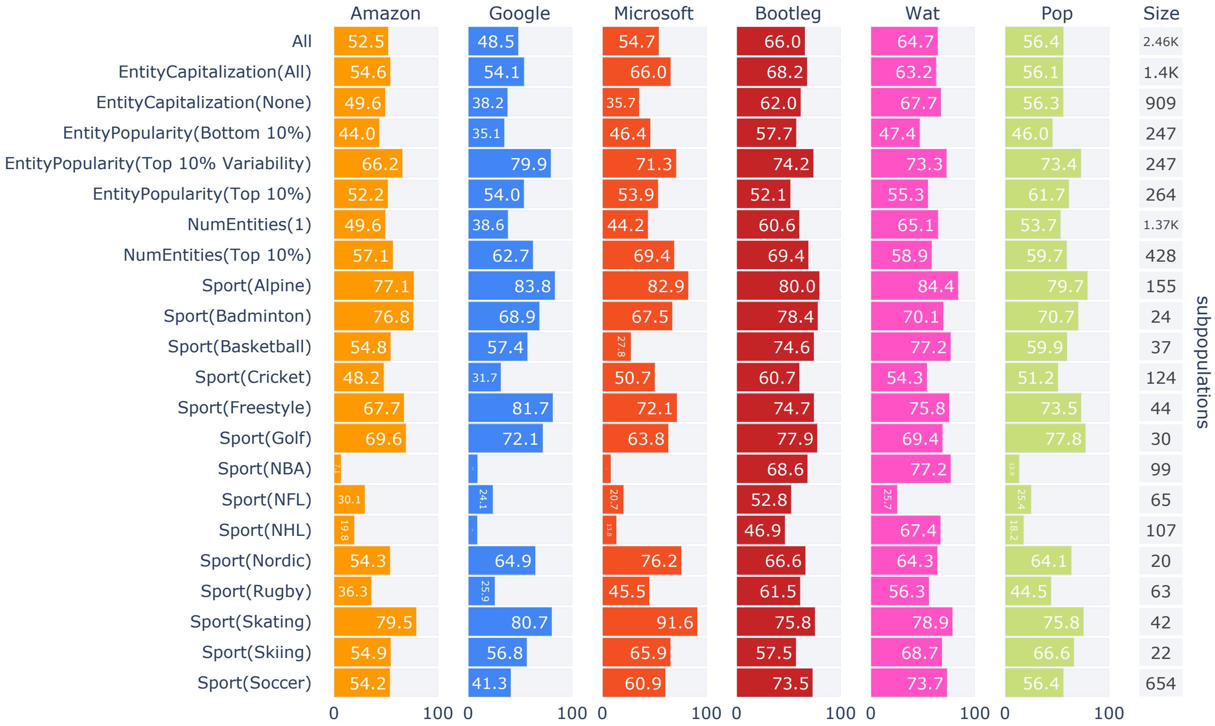


Figure 4: Robustness Report (Goel et al., 2021b) for NEL on AIDA, measuring Macro-F1.

are capitalized to 38.2% on sentences where none are capitalized. Similarly, MICROSOFT degrades from 66.0% to 35.7%. This suggests that mention extraction in these models is capitalization sensitive. In contrast, AMAZON, BOOTLEG and WAT appear insensitive to capitalization artifacts.

Performance on topical entities. Interestingly, all models struggle on some topics, e.g. on NHL examples, all models degrade significantly, with WAT outperforming others by 20%+. GOOGLE and MICROSOFT display strong performance on some topics, e.g., GOOGLE on alpine sports (83.8%) and MICROSOFT on skating (91.6%).

Popularity heuristic outperforms commercial systems. Somewhat surprisingly, POP outperforms all commercial systems by 1.7%. In fact, we note that the pattern of errors for POP is very similar to those of the commercial systems, e.g., performing poorly on NBA, NFL and NHL slices. This suggests that commercial systems sidestep the difficult problem of disambiguating ambiguous entities in favor of returning the more popular answer. Similar to WIKIPEDIA, GOOGLE performs best among commercial systems on examples with globally popular entities (top 10% entity popularity).

Our results suggest that state-of-the-art academic systems outperform commercial APIs for NEL.

Next, we explore whether it is possible to simply "patch" an off-the-shelf NEL model for a specific downstream use case. Standard methods for designing models with desired capabilities require technical expertise to engineer the architecture and features. As these skills are out of reach for many organizations and individuals, we consider patching models where they are treated as a black-box.

We provide a proof-of-concept that we can use data engineering to patch a model. For our grounding use case, we consider the scenario where the NEL model will be used as part of a sports question-answering (QA) system that uses a knowledge graph (KG) to answer questions. For example, given the question "When did England last win the FIFA world cup?", we would want the NEL model to resolve the *metonymic* mention "England" to the English national football team, and not the country. This makes it easy for the QA model to answer the question using the "winner" KG-relationship to the 1966 FIFA World Cup, which applies only to the team and not the country.

3.3 Predicting the Wrong Granularity

Our off-the-shelf analysis revealed that all models struggle on sport-related subpopulations of AIDA. For instance, Bootleg is biased towards predicting countries instead of sport teams, even with strong contextual cues. For example, in the sentence "...the years I spent as manager of the Republic of Ireland were the best years of my life", Bootleg predicts the country "Republic of Ireland" instead of the national sports team. In general, this makes it undesirable to directly use off-the-shelf in our sports QA system scenario.

We explore repurposing in a controlled environment using Bootleg, the best-performing off-the-shelf NEL model. We train a small model, called BootlegSport, over a Wikipedia subset consisting only of sentences with mentions referring to both countries and national sport teams. We define a subpopulation, *strong-sport-cues*, as mentions directly preceded by a highly correlated sport team cue[3]. Examining *strong-sport-cues* reveals two insights into BootlegSport's behavior:

1. BootlegSport misses some strong sport-relevant textual cues. In this subpopulation, 5.8% examples are mispredicted as countries.

2. In this supopulation, an estimated 5.6% of mentions are *incorrectly* labeled as countries in Wikipedia. As Wikipedia is hand labeled by users, it contains some label noise.

In our use case, we want to guide BootlegSport to always predict a sport team over a country in sport-related sentences.

3.4 Repurposing with Weak Labeling

While there are some prior data engineering solutions to "model patching", including augmentation (Sennrich et al., 2015; Wei and Zou, 2019; Kaushik et al., 2019; Goel et al., 2021a), weak labeling (Ratner et al., 2017; Chen et al., 2020), and synthetic data generation (Murty et al., 2020), due to the noise in Wikipedia, we repurpose BootlegSport using weak labeling to modify training labels and correct for this noise. Our weak labeling technique works as follows: any existing mention from *strong-sport-cues* that is labeled as a country is relabeled as a national sports team for that country. We choose the national sport team to be consistent with other sport entities in the sentence. If there are none, we choose a random national sport team. While this may introduce noise, it allows us to guide BootlegSport to prefer sport teams over countries.

Subpop.	Gold Label	Pred. Label	Size (Off-The-Shelf → Patched)	
All	Country	Country	90885 → 90591	(↓)
		Team	201 → 254	(↑)
	Team	Country	216 → 161	(↓)
		Team	4057 → 4120	(↑)
Weak Sport Cues	Country	Country	15225 → 15139	(↓)
		Team	154 → 190	(↑)
	Team	Country	151 → 106	(↓)
		Team	3393 → 3447	(↑)

Table 1: BootlegSport prediction matrix before and after model patching. The weak sport cues subpopulation contains sentences with more generic sport related keywords.

Results. After performing weak labeling, we finetune BootlegSport over this modified dataset. As Wikipedia ground truth labels are noisy and do not reflect our goal of favoring sport teams in sport sentences, we examine the distribution of predictions before and after guiding. In Table 1 we see that our patched model shows an increased trend in predicting sport teams. Further, the patched BootlegSport model now only predicts countries in 4.0% of the *strong-sport-cues* subpopulation, a 30% relative reduction.

For examples where the gold entity is a sports team that BootlegSport predicts is a country, weak labeling improves absolute accuracy by 24.54%. Weak-labeling "shifts" probability mass from countries towards teams by 20% on these examples, and 1.8% overall across all examples where the gold entity is a sports team. It does so without "disturbing" probabilities on examples where the true answer is indeed a country, where the shift is only 0.07% towards teams.

4 Related Work

Identifying Errors. A key step in assessing off-the-shelf systems is *fine-grained evaluation*, to determine if a system exhibits undesirable behavior. Prior work on fine-grained evaluation in NEL (Rosales-Méndez et al., 2019) characterizes how to more consistently evaluate NEL models, with an analysis that focuses on academic systems. By contrast, we consider both academic and industrial off-the-shelf systems, and describe how to assess them in the context of a downstream

[3] We mine these textual cues by looking at the most commmon two-grams proceeding a national sport team in the training data. The result is phrases such as "scored against", "match against", and "defending champion".

use-case. We use Robustness Gym (Goel et al., 2021b), an open-source evaluation toolkit for performing the analysis, although other evaluation toolkits (Ribeiro et al., 2020; Morris et al., 2020) are possible to use, depending on the objective of the assessment.

Patching Errors. If a system is assessed to have some undesirable behavior, the next step is to correct its errors and repurpose it for use. The key challenge lies in how to correct these errors. Although similar to the related fields of domain adaptation (Wang and Deng, 2018) and transfer learning (Zhuang et al., 2020) where the goal is to transfer knowledge from a pretrained, source model to a related task in a potentially different domain, our work focuses on user-guided behavior correction when using a pretrained model on the same task.

For industrial NEL applications, Orr et al. (2020) describe how to use data management techniques such as augmentation (Sennrich et al., 2015; Wei and Zou, 2019; Kaushik et al., 2019; Goel et al., 2021a), weak supervision (Ratner et al., 2017), and slice-based learning (Chen et al., 2019) to correct underperforming, user-defined sub-populations of data. Focusing on image data Goel et al. (2021a) use domain translation models to generate synthetic augmentation data that improves underperforming subpopulations.

NEL. NEL has been a long standing problem in industrial and academic systems. Standard, pre-deep-learning approaches to NEL have been rule-based (Aberdeen et al., 1996), but in recent years, deep learning systems have become the new standard (see Mudgal et al. (2018) for an overview of deep learning approaches to NEL), often relying on contextual knowledge from language models such as BERT (Févry et al., 2020) for state-of-the-art performance. Despite strong benchmark performance, the long tail of NEL (Bernstein et al., 2012; Gomes, 2017) in industrial workloads has remained a challenge. Recent papers Orr et al. (2020); Wu et al. (2019) have begun to measure and improve performance on unseen entities, but it remains an open problem.

5 Conclusion

We studied the performance of off-the-shelf NEL models and how to repurpose them for a downstream use case. In line with prior work, we found that off-the-shelf models struggle to disambiguate rare entities. Using a sport QA system as a case study, we showed how to use a data engineering solution to patch a BOOTLEG model from mispredicting countries instead of sports teams. We hope that our study of data engineering to effectuate model behavior inspires future work in this direction.

Acknowledgements

CR gratefully acknowledges the support of NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); ONR under No. N000141712266 (Unifying Weak Supervision); the Moore Foundation, NXP, Xilinx, LETI-CEA, Intel, IBM, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, the Okawa Foundation, American Family Insurance, Google Cloud, Swiss Re, Total, the HAI-AWS Cloud Credits for Research program, the Salesforce AI Research grant and members of the Stanford DAWN project: Facebook, Google, and VMWare. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of NIH, ONR, or the U.S. Government.

References

John Aberdeen, John D Burger, David Day, Lynette Hirschman, David D Palmer, Patricia Robinson, and Marc Vilain. 1996. Mitre: Description of the alembic system as used in met. In *TIPSTER TEXT PROGRAM PHASE II: Proceedings of a Workshop held at Vienna, Virginia, May 6-8, 1996*, pages 461–462.

Amazon. Amazon comprehend api.

Michael S Bernstein, Jaime Teevan, Susan Dumais, Daniel Liebling, and Eric Horvitz. 2012. Direct answers for search queries in the long tail. In *SIGCHI*.

Mayee F. Chen, Daniel Y. Fu, Frederic Sala, Sen Wu, Ravi Teja Mullapudi, Fait Poms, Kayvon Fatahalian, and Christopher Ré. 2020. Train and you'll miss it: Interactive model iteration with weak supervision and pre-trained embeddings. *arXiv preprint arXiv:2006.15168*.

Vincent Chen, Sen Wu, Alexander J Ratner, Jen Weng, and Christopher Ré. 2019. Slice-based learning: A programming model for residual learning in critical data slices. In *Advances in neural information processing systems*, pages 9392–9402.

P. Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). *ArXiv*, abs/1006.3498.

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

Abbas Ghaddar and Philippe Langlais. 2017. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422.

Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. 2021a. Model patching: Closing the subgroup performance gap with data augmentation. In *The International Conference on Learning Representations (ICLR)*.

Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. 2021b. Robustness gym: Unifying the nlp evaluation landscape.

Ben Gomes. 2017. Our latest quality improvements for search. `https://blog.google/products/search/our-latest-quality-improvements-search/`.

Google. Google cloud natural language api.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Johannes M. van Hulst, F. Hasibi, K. Dercksen, K. Balog, and A. D. Vries. 2020. Rel: An entity linker standing on the shoulders of giants. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Yuzhe Jin, Emre Kıcıman, Kuansan Wang, and Ricky Loynd. 2014. Entity linking at the tail: sparse signals, unknown entities, and phrase models. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 453–462.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Microsoft. Microsoft text analytics api.

John X Morris, Eli Lifland, Jin Yong Yoo, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks in natural language processing. *arXiv preprint arXiv:2005.05909*.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34.

Shikhar Murty, Pang Wei Koh, and Percy Liang. 2020. Expbert: Representation engineering with natural language explanations. *arXiv preprint arXiv:2005.01932*.

L. Orr, Megan Leszczynski, Simran Arora, Sen Wu, N. Guha, Xiao Ling, and C. Ré. 2020. Bootleg: Chasing the tail with self-supervised named entity disambiguation. *CIDR*.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Francesco Piccinno and P. Ferragina. 2014. From tagme to wat: a new entity annotator. In *ERD '14*.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Association for Computational Linguistics (ACL)*.

Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. 2019. Fine-grained evaluation for entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 718–727.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2020. Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27:443–460.

Nadine Steinmetz, Magnus Knuth, and Harald Sack. 2013. Statistical analyses of named entity disambiguation benchmarks. In *NLP-DBPEDIA@ ISWC*.

Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

Jason W Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.