Proceedings of the Federated Conference on Computer Science and Information Systems pp. 189–192
DOI: 10.15439/2020F24
ISSN 2300-5963 ACSIS, Vol. 21

Named Entity Recognition and Named Entity on Esports Contents

Ziyu Liu*, Yifan Leng†, Meiqi Wang‡ and Congzhu Lin§
Michtom School of Computer Science, Brandeis University
Waltham, Massachusetts, USA
Email: *ziyuliu@brandeis.edu, †yifanleng@brandeis.edu, ‡meiqw@brandeis.edu, §linc@brandeis.edu

***Abstract*—We built a named entity recognition system on Esports News. We established an ontology for Esports-related entities, collected and annotated corpus from 80 articles on four different Esports titles. We also trained a CRF and a BERT-based entity recognizer, built a basic DOTA2 knowledge base, and an entity linker that links mentions of entities to articles in Liquipedia (the Esports Wikipedia), and a naive web app which serves as a demo of this entire proof-of-concept system. We achieved an over 61% overall entity-level F1-score on the test set for the NER task.**

I. Introduction and Related Works

NAMED entity recognition (NER) has been a popular topic within the NLP area. As defined in the MUC7[1] definition, the goal of the task is to find unique identifiers of entities (organizations, persons, locations), times and quantities, and to identify all of those expressions in each text in the test set, and to categorize them.

Entity linking[2](EL) is a new task on the computation linguistics field. The goal of it is, besides identifying mentions of identities within the given text, linking them to the most suitable entry within a reference knowledge base.

NER and EL both address the lexical ambiguity of language and play important roles towards the broader goal of the NLP research: the automatic understanding of natural languages. While the problem of NER/EL tasks on formal text, like news, there is almost no study about NER/EL on texts about new emerging topics, such as Esports news. We are aware that some existing NER works[3], [4], [5] covered the corpus in the sports domain. However, arguably, Esports news are generally more informal, shorter and having a broader types of entities(e.g. virtual characters, users' online ids, etc.). And obviously we would need a different ontology to address these differences.

Another fact motivates our work is that recent years have witnessed a booming Esports industry. It had an estimated market worth of 138 billion US dollars in 2018, according to market research firm Newzoo. Esports news websites such as JoinDota.com, dotesports.com, liquipedia.net, have created a significant amount of high-quality news content covering matches results, transferring, and commercial insights on a variety of popular Esports titles.

Reliable NER/EL system on Esports contents could serve as essential parts in larger real-world NLP systems like automatic Esports news taggers, Esports match result prediction systems, or players/teams popularity analyzers. Moreover, there is no doubt that those systems have great potential in both academic and economic values.

In this paper, we established an ontology for Esports-related entities, collected and annotated corpus from 80 articles on 4 different Esports titles, trained CRF[6] and BERT-based[7] entity recognizer, built a basic DOTA2 knowledge base, an entity linker that links mentions to articles in Liquipedia, and an end-to-end web app which serves as a demo of this entire proof-of-conecpt system.

The rest of the paper is organized as follows. Section II describes the process of collecting corpus. In section III we introduce the ontology we set for the system and explain its rationale; section IV discusses the models and feature sets we used for the NER task; section V shows how we built the DOTA2 knowledge base; section VI explains how does the entity linking system works; and section VIII illustrates how the web app is built. Section VII reports the setting and results of our experiments on NER task and shows perceptive results of the entity linker. And finally, we conclude our project and propose valuable future works on the topic in section IX.

II. Corpus Collection

We limited our scope to four popular games: DOTA2[1], League of Legends[2], CS:GO[3], and Overwatch[4].

Our first attempt was collecting Esports data from Twitter by searching game names. Twitter has sufficient text data, and it is easy to retrieve tweets with Twitter APIs. However, there were two significant issues that discouraged us from using Twitter as the primary data source: 1. Although Twitter has an abundance of data, Esports-related entities are relatively sparse in tweets. 2. Twitter poses rate limits on accessing tweets and other information, e.g., searching tweets is limited to 180 requests per window, where each window is 15 minutes in length[5]. Crawling a large amount of data would be inefficient.

We then decided to utilize Esports news websites (e.g., dotesports.com). These websites are frequently updated by professional editors and contain more condensed information about tournaments, player transfers, and more.

[1]http://blog.dota2.com/
[2]https://signup.na.leagueoflegends.com
[3]https://blog.counter-strike.net/
[4]https://playoverwatch.com
[5]https://developer.twitter.com/en/docs/basics/rate-limits

We handpicked 25 articles for each game, where 20 were used for training/development set, five were held out as the test set. Each article contains 300 - 800 words and has at least 5 Esports entities.

III. Ontology

A. The First Attempt

Our original ontology contained six tags: GAME (game), TOURN (tournament), ORG (organization), PLAYER (player), PERF (performance), and SPONS (sponsor), defined as below:

- GAME: The Esports title.
- TOURN: An Esports event or league.
- ORG: The team in which name players play for.
- PLAYER: Individuals who play and compete on the game as a career (in other words, "pro player").
- PERF: Any comments on the player/team's performance on a certain game, a series(set of games).
- SPONS: Third-party sponsor of the event/organization.

B. Refined Ontology

After annotated all articles, we ran our baseline CRF averaged perceptron model and reached over a 0.50 F1 score on all entities except PERF and SPONS. We had zeroes on PERF. PERF contained long text spans (e.g. "He [dominated the DOTA Summit 11 Minor] with iG"). Besides, PERF was relatively difficult to define: it can be any comments on players or teams on a certain game or a series. The annotator agreement was low and might have impacted the performance of PERF. SPONS was absent in training articles, and therefore our baseline model did not tag any SPONS entity in the test set.

We later dropped PERF and SPONS, and added another entity called "AVATAR". AVATAR represents a player's role in the game, and it is an essential part of the gameplay. In DOTA2, League of Legends, and Overwatch, players each control their own characters. Each character has different abilities and functions. CS:GO does not have explicitly defined characters, but items in the game can define the roles and functions. For example, a support is generally the person carrying the flashbangs, molotovs, grenades, etc.[6].

Our refined ontology contained five kinds of entities: GAME, TOURN, ORG, PLAYER, and AVATAR, listed below:

- GAME: The Esports title.
- TOURN: An Esports event or league.
- ORG: The team in which name players play for.
- PLAYER: Individuals who play and compete on the game as a career (in other words, "pro players").
- AVATAR: The character that a player controls. In CS:GO, it is the weapon/items a player uses.

[6]https://www.pinnacle.com/en/esports-hub/betting-articles/cs-go/a-guide-to-csgo-role/ml2jx57tyd6bxr7z

IV. NER Models

We tried two different NER models on this task.

As for the CRF model, we used the averaged perceptron in `CRFSuite` package. We did some ablation tests to determine the best feature set to use and at last the feature set we used are `Bias`, `Token`, `Uppercase`, `Titlecase`, `Digit`, `Punctuation`, and `WordShape`. `BrownCluster` and `WordVector` are discarded as they turned out to hinder the model's performance. We believe that they should be useful if those representations are trained on Esports-related corpus.

For the BERT model, we used the open source software on https://github.com/kyzhouhzau/BERT-NER with some modification to make it work with our ontology. All of the parameters are remained as default.

On the web app backend, we choose to use the CRF model, as it requires much less computation resources.

V. Knowledge Base Building

Entity requires a well-structured knowledge base as target. Under common scenarios, the target knowledge base is usually built based on Wikipedia[7]. However, although there are surely some articles on Esports entities, Wikipedia is far from comprehensive. Instead we would use Liquipedia[8], one of the biggest Esports wiki sites as the source of our knowledge base.

Undoubtedly, Liquipedia is a comprehensive and reliable source of information, but by choosing it as our target, it also introduces several challenges:

- Poorly-documented-and-implemented APIs. The MediaWiki APIs that Liquipedia provided are not well-documented. And most importantly, many critical actions, like dumping or parsing are not supported or implemented. To address this, we have to write our own crawler to retrieve and parse the document tree in order to extract useful, structured information.
- Inconsistency across sub-sites. Liquipedia is formed of several subsites, e.g. https://liquipedia.net/dota2/, https://liquipedia.net/starcraft2 and https://liquipedia.net/overwatch/. These sub-sites, although looks similar, seem to have slightly different front-end coding. And, as these are different Esports games, these sub-sites are organized differently, inherently. Therefore, it is hard to write a crawler which can easily build a knowledge base that contains all information for all Esports titles. For this reason, we currently only built a knowledge base for DOTA2.
- Access frequency limitation. This is a common practice for most modern websites, that an IP will be banned for a certain period of time, if it is sending requests to the server too frequently. It turns our that this issue is relatively easy to tackle, by simply putting `sleep(2)` on each request.

[7]https://en.wikipedia.org/

[8]https://liquipedia.net/

A. Crawler

To build the crawler, we used `beautifulsoup` as our HTML parser, and we collected all information on teams(organizations), players, tournaments and heroes, then organized and saved them into json files. We also considered putting them into SQL-based database to enable more query functions. However, considering we will only mostly doing key-value searching/ranking operations, and the total data size is only about 400kB, we decided to store them just as json file.

An example tournament entry would look like:

```
"tier": "Major",
"name": "China DOTA2 Professional League
Season 1",
"dates": "Oct 17, 2019 - Mar 1, 2020",
"prize_pool": 212690,
"teams": "10",
"host_location": "China",
"event_location": "Online"
```

VI. Entity Linking

The actual entity linking process is initiated after the entities in the given text are recognized. To determine which entry in the knowledge base should be returned, we query the knowledge base using the text as key, under the recognized entity type. If successful, an entry containing all related information will be returned and used in the next step (in our system, being rendered on the web page).

A. Query Handling

When a query string is passed to the knowledge base, the actual key is returned based on Algorithm1. Inside which, the candidate key set ξ is built when the system is initialized, by combining all `names` and `aliases` in the JSON files.

Algorithm 1 Get matching key

Require: s: query string, ξ : candidate key set
  if $s \in \xi$ **then**
    return s
  end if
  for $k \in \xi$ **do**
    if s is substring of k **then**
      return k
    else
      get close match of s, $s' \in \xi$
    end if
  end for

VII. Evaluation

This section reports our experiment results with different NER models and web app demo screenshots.

For NER tasks, we use entity-level precision/recall/F1 as our metrics, which are calculated based on the whether the prediction for an entity matches perfectly with the true entity start/end labels.

A. CRF Averaged Perceptron

TABLE I
AP WITH ALL FEATURES

Type	Prec	Rec	F1
ALL	54.83%	52.16%	53.46%
AVATAR	59.57%	35.44%	44.44%
GAME	64.29%	100.00%	78.26%
ORG	69.73%	53.75%	60.71%
PLAYER	44.44%	53.01%	48.35%
TOURN	38.71%	61.54%	47.52%

TABLE II
REMOVE BROWN CLUSTER AND WORD VECTOR

Type	Prec	Rec	F1
ALL	59.37%	52.91%	55.95%
AVATAR	47.76%	40.51%	43.84%
GAME	66.67%	88.89%	76.19%
ORG	79.55%	58.33%	67.31%
PLAYER	44.71%	45.78%	45.24%
TOURN	52%	66.67%	58.43%

TABLE III
BEST FEATURE SET*

Type	Precision	Recall	F1
ALL	62.24%	55.35%	58.59%
AVATAR	57.81%	46.84%	51.75%
GAME	88.89%	88.89%	88.89%
ORG	80.00%	60.00%	68.57
PLAYER	46.71%	46.99%	46.85%
TOURN	52.83%	71.79%	60.87%

*Best feature set: Bias, Token, UpperCase, Titlecase, Digit, Punctuation, WordShape

B. BERT NER

TABLE IV
BEST RESULT

Type	Precision	Recall	F1
ALL	62.35%	69.05%	61.22%
AVATAR	50.00%	2.56%	4.88%
GAME	30.00%	37.50%	33.33%
ORG	64.73%	87.91%	74.56%
PLAYER	71.52%	81.38%	76.13%
TOURN	41.03%	55.17%	47.06%

We can see that although the BERT model outperforms CRF-AP in terms of overall F1-score and on ORG and PLAYER. However, it falls short on GAME, TOURN and especially, AVATAR. We believed the much lower recall/F1 score on AVATAR, compared to CRF model, is caused by the lack of model fine-tuning for the task.

VIII. Web Application

To more conveniently assess the performance of the trained model, we built and deployed a web application [9] on

[9]http://lengyifan.pythonanywhere.com/

`PythonAnywhere`. The web application uses the trained CRF model to perform tagging on the text snippet. Five test documents were selected from the database that shows the named entities predicted by the model once clicked. The user can also input a text snippet or a URL in the search bar, and the text body will be extracted to perform named entity tagging on.

The tagged named entity is re-directed to a page that shows its related information in Liquipedia with a URL. Different mentions will have the same URL in this information section if they are the same entity (e.g "Invictus Gaming" and its alias "IG" or "iG"), suggesting successful entity . To inspect the mechanism the model uses to predict the label, we displayed the sentences in the training docs where the tagged entity is annotated. It facilitates understanding and selecting the features in the training and tagging stage.

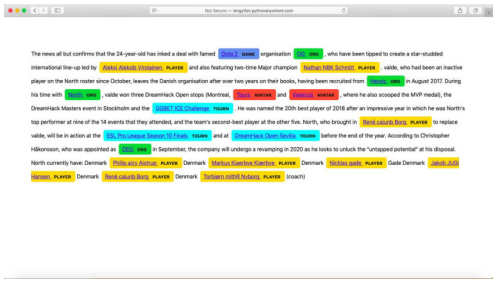

Fig. 1. Named-Entity Tagging Page

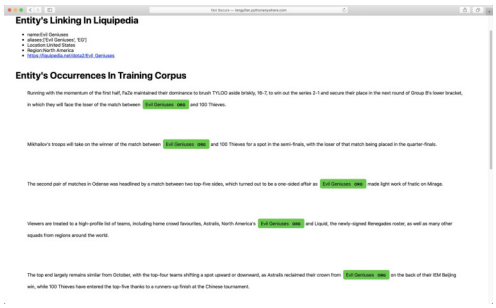

Fig. 2. Named-Entity Detail Page

Figure 1 shows a piece of sample marked input text produced by the entity tagger with each named entity colored separately. Figure 2 shows the detail page of the recognized entity in Figure 1. With our linking algorithm, different aliases (such as "Evil Geniuses" and "EG") are mapped to their unique identity which is an entry in Liquipedia. We show this Liquipedia entry as its identity along with all of the entity's occurrences in the training corpus on this detail page.

IX. Conclusion and Future Works

In this project, we collected and annotated corpus from Esports news using the ontology set by ourselves, conducted NER experiment using CRF and BERT models on the test data set, and built an end-to-end Esports entity Liquipedia system which is capable of recognizing Esports players, teams and tournaments from texts. Although the system did not yield a satisfying result for AVATAR and GAME entities, we still managed to achieve 61.22% overall F1 score for the NER task using the BERT model, 58.59% overall F1 score using the CRF model.

This paper should serve as a starting point to combine NLP techniques and new emerging fields like Esports. As for future work, we consider these directions as the most meaningful ones:

- Better, finer-tuned NER model. Our CRF and BERT models are not fine-tuned; and as the the Recall and F1 score on AVATAR is abnormally low for BERT, we think there should be a large room of future improvement.
- Refining the knowledge base query algorithm to be ranking-based. Current system would be very likely to return false results when two teams has the same aliases. This could be undermined if we could let the system do ranking based on other information in the given text.
- Building a knowledge base contains more Esports titles so that our system can work on more games. This would be done easily if Liquipedia could help to provide an article dump api.
- Building a corpus on non-formal sources like social media.
- Supporting Esports contents in languages other than English. We found that there are some higher-quality Chinese and Russian corpus and would recommend navigating towards this direction.

References

[1] N. Chinchor and P. Robinson, "Muc-7 named entity task definition," in *Proceedings of the 7th Conference on Message Understanding*, vol. 29, 1997, pp. 1–21.

[2] D. Rao, P. McNamee, and M. Dredze, "Entity linking: Finding extracted entities in a knowledge base," in *Multi-source, multilingual information extraction and summarization*. Springer, 2013, pp. 93–115.

[3] T. Yao, W. Ding, and G. Erbach, "Chiners: a chinese named entity recognition system for the sports domain," in *Proceedings of the second SIGHAN workshop on Chinese language processing*, 2003, pp. 55–62.

[4] C.-K. Lee and M.-G. Jang, "Named entity recognition with structural svms and pegasos algorithm," *Korean Journal of Cognitive Science*, vol. 21, no. 4, pp. 655–667, 2010.

[5] X. Seti, A. Wumaier, T. Yibulayin, D. Paerhati, L. Wang, and A. Saimaiti, "Named-entity recognition in sports field based on a character-level graph convolutional network," *Information*, vol. 11, no. 1, p. 30, 2020.

[6] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.