Optimizing REL

Erik Tjong Kim Sang
Netherlands eScience Center

January 16, 2023

1 Introduction

REL (Radboud Entity Linker) [9] is software for disambiguating names in texts. Names, also called named entities or entities, can have different meanings. For example Ajax can refer to a Greek hero, a soccer club, a programming concept, a detergent brand and much more. An entity linker like REL uses the context of a name in a text to find the most relevant Wikipedia page for the name.

Several recent work has adapted and optimized REL. Aydin et al. [2] adapted REL for the task of finding funding information in text and replaced the standard Flair entity detector by Bert. Joko et al. [4] extended REL to processing conversations. Kamphuis [5] optimized REL with the goal to quickly process a large number of documents. This report investigates the effects of replacing various parts of REL by other software and evaluates the effects of these changes.

2 Evaluation of REL

REL performs two tasks: 1. mention detection (MD), finding names in text, and 2. entity disambiguation (ED), linking the detected names to the right Wikipedia article. The combined task, mention detection followed by entity disambiguation is called entity linking (EL). These three tasks are being evaluated separately. ED is evaluated by its own while presuming a perfectly performing preceding MD task.

There are two ways to evaluate the performance of REL: 1. with the script `scripts/efficiency_test.py`, and 2. with the external service Gerbil. This section discusses both options.

2.1 Evaluating REL with the script `efficiency_test.py`

The script `scripts/efficiency_test.py` can be used for evaluating the speed of REL. The script needs to be changed before it can be run:

1. download REL: `git clone https://github.com/informagi/REL.git`

2. `cd REL`
3. edit script: `edit scripts/efficiency_test.py` and make these changes:
   (a) add local value of `base_url` variable (lines 8 and 59)
   to prevent the error: `No such file ...: .../persons.txt`
   (b) set value of `server` variable to `False` (line 14)
   to prevent the error `Connection timed out`
   (c) set value of `flair.device` variable to `cpu` (line 61)
   to prevent the error `Found no NVIDIA driver on your system`
4. install software: `pip install .`
5. run evaluation: `python scripts/efficiency_test.py`

We will evaluate entity linking with respect to two Wikipedia versions, one from 2014 and one from 2019, just like Van Hulst et al. [9]. The Wikipedia version can be set in `scripts/efficiency_test.py`. Running the script can take up to six minutes. It reports the times in seconds taken by mention detection (MD) and entity disambiguation (ED), for example: MD took: 226.0 (seconds) and ED took: 93.3.

The script can be instructed to communicate with the REL entity linking server. This requires the following additional steps:

1. edit script: `edit scripts/efficiency_test.py` and make these changes:
   (a) set value of `server` variable to `True` (line 14)
   (b) insert the correct host name and the correct port (5555) in `requests_post` (line 43)
   to prevent the error `Connection timed out`
2. install software: `pip install .`
3. start entity linking server:
   `python3 REL/server.py --ed-model ed-wiki-2019 data wiki_2019`[1]
4. run evaluation: `python scripts/efficiency_test.py`

Running `scripts/efficiency_test.py` while using the entity linking server can take up to 14 minutes. The script will not report the used running times but it will show the names that were found with their associated Wikipedia lemmas.

Extra code has been written to estimate the precision and recall in the data processed by `efficiency_test.py` (`evaluate_predictions.py`). The code compares predicted entities with required entities from the gold standard without considering the position of the entities in the document (the D2KB method used by Gerbil). If a match is found both for the entity and its Wikipedia link, MD recall and the EL recall are increased. If the match only concerns the predicted entity but not the Wikipedia link, only the MD recall score is increased. When all entities have been processed, precision scores are computed for MD and EL. The results for the two Wikipedia data versions can be found in Table 1. The

[1] Add the option –process_sentences to get the default behaviour of the server.

	Mention Detection				Entity Linking			
Wiki	Time	Prec.	Rec.	F1	Time	Prec.	Rec.	F1
2014	381 sec.	97.4%	61.4%	75.3%	9 sec.	70.3%	44.3%	54.4%
2019	282 sec.	97.4%	61.8%	75.7%	10 sec.	65.8%	41.7%	51.1%
2014	server	97.6%	61.5%	75.4%	server	70.4%	44.4%	54.4%
2019	server	97.7%	61.9%	75.8%	server	65.9%	41.8%	51.1%

Table 1: Performance scores of REL for mention detection (MD) and entity linking (EL=MD+ED), processing 50 documents, estimated for two Wikipedia years, with `scripts/efficiency_test.py`, without (top two lines) and with using an annotation server (bottom two lines).

reported F1 scores are micro F1 scores. Since the scores obtained by using the server are similar to those obtained without the server, we will use the faster serverless method for this type of evaluation for the rest of this study.

2.2 Evaluating REL with Gerbil

The second evaluation method relies on the online evaluator Gerbil [8]. This requires running two servers: the entity linking server, like in the previous evaluation method, and the Gerbil middleware:

1. `cd REL`
2. `python3 REL/server.py --ed-model ed-wiki-2019 data wiki_2019`[1]
3. `cd REL/scripts/gerbil_middleware`
4. set Jena version to 2.11.1 in pom.xml line 79: `edit pom.xml`
   to prevent the error: java.lang.NoClassDefFoundError: com/hp/hpl/jena/reasoner/Reasoner
5. replace initial spaces by tabs (lines 4, 7 and 10): `edit Makefile`
6. run make: `make`
7. `mvn clean -Dmaven.tomcat.port=1235 tomcat:run`

An experiment can be started on Gerbil via the web form `https://gerbil.aksw.org/gerbil/config`. The following data need to be filled in the form:

- Experiment type: A2KB
- Matching: Ma - strong annotation match
- Annotator Name: any string
- Annotator URI: http://HOSTNAME:1235/gerbil-spotWrapNifWS4Test/myalgorithm , with HOSTNAME replaced by the name of the host where the Gerbil middleware runs
- Dataset: AIDA/CoNLL–Test B

Next click on the disclaimer check box and press the button *Run Experiment* to start the evaluation. The results will become available at the Gerbil website (see Table 3).

Flair output		Bert output	
type	phrase	type	phrase
PER	Shu Kamo	B-PER	Shu
		I-PER	Ka
		I-PER	##mo

Figure 1: Example of the output format differences between Flair and Bert: entities in the output of Bert can be split both in several words as word-internally while the entities in the Flair output are complete

3 Optimization

Currently (2022) mention detection in REL is performed by software called Flair [1]. We will start with replacing this software by Bert [6]. This will both have an effect on the performance in the MD task, measured in precision, recall and F1, and on the efficiency of the MD task, measured in seconds per number of processed documents. There are two ways to evaluate these: with `scripts/efficiency_test.py` and with the web service Gerbil (see previous section). We evaluate entity linking with respect to one Wikipedia version, the one from the year 2019.

3.1 Replacing Flair with Bert

The present software of REL needs to be modified to be able to run with Bert:

1. in `scripts/efficiency_test.py`:
   - load the new Bert module:
     `from REL.ner.bert_wrapper import load_bert_ner`
   - use the Bert NER tagger:
     `tagger_ner = load_bert_ner("dslim/bert-base-NER")`
2. in `REL/mention_detection.py` (lines 151-155) use Bert entity format:
   - `entity["word"],`
     replaces: `entity.text,`
   - `entity["start"],`
     replaces: `entity.start_position,`
   - `entity["end"],`
     replaces: `entity.end_position,`
   - `entity["score"],`
     replaces: `entity.score,`
   - `entity["entity"],`
     replaces: `entity.tag,`
3. change the type of `float32` to `float` in the output of Bert in `REL/server.py` to avoid the error: Object of type float32 is not JSON serializable

	Mention Detection				Entity Linking			
NER	Time	Prec.	Rec.	F1	Time	Prec.	Rec.	F1
Flair	203 s	**97.4%**	61.9%	**75.7%**	14 s	**65.7%**	**41.7%**	**51.0**%
Bert large	125 s	91.6%	**62.9%**	74.6%	**12 s**	60.7%	**41.7%**	49.4%
Bert base	**47 s**	89.6%	60.5%	72.2%	13 s	59.4%	40.1%	47.9%

Table 2: Performances measured by `scripts/efficiency_test.py` for mention detection and entity linking with two different NER software: Flair and Bert, for one Wikipedia version (2019).

4. combine the entity parts in the output of Bert in `REL/mention_detection.py` (see Figure 1)
5. check the format of the Bert output in `REL/server.py`
6. then reinstall REL: `pip install .`

There are several Bert versions, we tested Bert large NER (cased) [3][2] and its smaller version Bert base NER[3]. The results of the evaluation can be found in Table 2. Precision, recall and F1 scores have been estimated. Bert large performs slightly worse than Flair. Bert base does worse than Bert large but it is almost three times as fast.

3.2 Evaluating Bert with Gerbil

The second evaluation method for REL relies on the online evaluator Gerbil [8]. This requires running two servers: the entity linking server, like in the previous evaluation method (step 3) and the Gerbil middleware (on the machine gem.cs.ru.nl):

1. `cd REL`
2. make sure Bert output is in same format as Flair output (elements 3 and 4 of entity data format are swapped)
3. `python3 REL/server.py --ed-model ed-wiki-2019 data wiki_2019`
4. `cd REL/scripts/gerbil_middleware`
5. `mvn clean -Dmaven.tomcat.port=1235 tomcat:run`

We used Gerbil with the parameter strong matching and the data set AIDA /ConNLL Test B. The evaluation results can be found in Table 3. As could have been expected, the numbers for Flair are almost identical to the evaluation scores of Van Hulst et al. [9] (Table 1, row REL (2019), column AIDA-B: 78.6, 80.5). The differences in the scores of Table 2 and Table 3 are caused by the fact that they are based on different test sets. The longest sentence in the Gerbil test dataset contains 124 tokens so the processing length restriction of Bert does not have an influence on these results. The scores of Bert are a lot lower than thos of Flair, it is unclear why.

[2]https://huggingface.co/dslim/bert-large-NER
[3]https://huggingface.co/dslim/bert-base-NER

NER	InKB Macro F1	InKB Micro F1	Experiment
Flair	**78.3%**	**80.3%**	202301050003
Bert large	72.0%	73.6%	202301050004
Bert base	70.9%	72.9%	202301050005

Table 3: Performances measured by Gerbil on entity linking with two different NER software: Flair and Bert, for one Wikipedia version (2019). The experiment id in the right column links to the page with the full evaluation results at the Gerbil website.

3.3 Error analysis

Figure 2 contains an example text from the Gerbil evaluation, processed by both Flair and Bert large (cased). On the basis of this text we can perform an error analysis to get more insight in the errors that Bert made for mention detection:

1. Bert made several strange errors: parts of words were identified as entities while other parts were not marked, for example *La Plagne* on line 12[4]
2. Bert had difficulty processing the initial line of the text which was entirely written in capital characters. It missed two of the three entities there.
3. Bert made different errors for team names like *Italy I* than Flair. Flair would identify the country names but ignore the team numbers. Bert identified both, but as separate entities.

The mistakes of the two system seem to be similar and not much can be concluded from the differences. We could try uncased Bert to avoid the errors observed in the first line of the test text. Since we want to make the system faster, we could also try to run Bert on entire documents.

3.4 Applying Bert to documents

Before REL applies Flair for identifying names in texts, the texts are first divided in sentences. This approach seems to harm the performance of Bert, which apparently needs more text context than just a sentence to correctly identify names. Thus we evaluated Bert while applied to complete documents. Before evaluation, all word internal entities were removed (see section 3.3, list point 1). The results can be found in Table 4. Processing documents (rows marked with *doc*) takes less time than processing sentences (rows marked with *sent*) but the performance (F1) drops. Since documents are longer than sentences, we have run into the maximum text size limit of Bert: 6% of the `scripts/efficiency_test.py` test documents and 6% of the Gerbil test documents contain more than 512 tokens, see Figure 3 for an example. By breaking

[4]The spit word tagging seems to be specific for this variant of Bert (cased) and was not observed in other variants, in particular not in uncased Bert.

Flair

BOBSLEIGH-SHIMER PILOTS USA III TO SURPRISE WIN. IGLS, Austria 1996-12-07 Brian Shimer piloted USA III to a surprise victory in a World Cup two-man bobsleigh race on Saturday. Lying fifth after the first run, Shimer and breakman Randy Jones delivered a near-perfect second trip down the 1976 Olympic course for an aggregate time of one minute 45.91 seconds. First run leaders Guenther Huber and breakman Antonio Tartaglia in the Italy I sleigh finished second two-hundredths of a second behind the Americans. Canada I, represented by Pierre Lueders and breakman Dave MacEachern, completed the third World cup event of the winter a further one-hundredth of a second behind the Italians. The Canadians, winners of the opening two events in Altenberg, Germany, and La Plagne, France, increased their lead in the World Cup standings. They have 104 points, 15 ahead of USA I's Jim Herberich and breakman Garrett Hines who managed only 10th place on Saturday.

Bert

BOBSLEIGH-SHIMER PILOTS USA III TO SURPRISE WIN. IGLS, Austria 1996-12-07 Brian Shimer piloted USA III to a surprise victory in a World Cup two-man bobsleigh race on Saturday. Lying fifth after the first run, Shimer and breakman Randy Jones delivered a near-perfect second trip down the 1976 Olympic course for an aggregate time of one minute 45.91 seconds. First run leaders Guenther Huber and breakman Antonio Tartaglia in the Italy I sleigh finished second two-hundredths of a second behind the Americans. Canada I, represented by Pierre Lueders and breakman Dave MacEachern, completed the third World cup event of the winter a further one-hundredth of a second behind the Italians. The Canadians, winners of the opening two events in Altenberg, Germany, and La Plagne, France, increased their lead in the World Cup standings. They have 104 points, 15 ahead of USA I's Jim Herberich and breakman Garrett Hines who managed only 10th place on Saturday.

Figure 2: Example text (AIDA/CoNLL-Test B number 192) with entities recognized by Flair (above) and Bert large cased (below). Phrases colored green (Flair: 18; Bert: 16) were correctly identified as entities while phrases colored red (Flair: 9; Bert: 17) were incorrect or missed.

				Mention Detection				Entity Linking			
	NER			Time	Prec.	Rec.	F1	Time	Prec.	Rec.	F1
	Flair										
1	sen	-	cas	200 s	97.5%	62.0%	75.8%	6.8 s	65.7%	41.7%	51.0%
2	doc	-	cas	231 s	97.5%	61.8%	75.7%	7.2 s	65.3%	41.4%	50.7%
4	sen	bas	cas	49 s	94.0%	60.5%	73.6%	16 s	62.2%	40.0%	48.7%
5	sen	lar	cas	154 s	94.0%	63.0%	75.4%	190 s	62.1%	41.6%	49.9%
6	sen	bas	unc	45 s	93.9%	62.5%	75.1%	17 s	62.6%	41.7%	50.1%
7	sen	lar	unc	124 s	96.3%	64.0%	76.9%	13 s	63.7%	42.4%	50.9%
8	doc	bas	cas	32 s	90.1%	55.3%	68.6%	11 s	59.8%	36.7%	45.5%
9	doc	lar	cas	89 s	88.2%	58.7%	70.5%	42 s	58.6%	39.0%	46.8%
10	spl	bas	cas	32 s	91.5%	58.5%	71.3%	6.5 s	60.6%	38.8%	47.3%
11	spl	lar	cas	91 s	90.9%	61.8%	73.6%	6.8 s	60.4%	41.1%	48.9%
12	spl	bas	unc	31 s	93.5%	62.4%	74.8%	11 s	62.9%	42.0%	50.3%
13	spl	lar	unc	87 s	96.6%	**64.1%**	**77.1%**	6.8 s	64.3%	**42.7%**	**51.3%**
14	spl	mul[4]	cas	33 s	97.7%	57.2%	72.2%	7.7 s	65.0%	38.0%	48.0%
15	sen	mul	cas	44 s	97.0%	57.2%	72.0%	8.2 s	63.6%	37.5%	47.2%

Table 4: Performances measured by `scripts/efficiency_test.py` for mention detection and entity linking with different NER software: Flair and Bert, comparing sentence processing (sen) vs document processing (doc) vs split document processing (spl: maximum 500 tokens, longer documents are split) and cased (cas) vs uncased (unc) processing. We tested three versions of Bert: large (lar), base (bas) and multi-lingual (mul). No uncased multi-lingual pretrained named entity recognition model was available for testing.

documents in smaller parts, the performance can be improved slightly (Table 4, rows marked with *split*). We use the same splitting strategy as Tigunova et al. [7]: split documents at sentence boundaries. Only when a sentence is found that is longer than the maximum document size, splits are made on other places than sentence boundaries. We use tokens as determined by Bert, which includes splitting unknown words in known tokens. We performed a number of tests to determine the optimal maximum document size (see Figure 3.4) and chose a maximum document size of 500 tokens, which corresponded with high F1 scores for both Bert base uncased split and Bert large uncased split.

Bert performs better when running as uncased (the default is cased), see Table 4, rows marked with *uncased*. This required adaption of the results of mention detection to prevent uncased entities (*japan* instead of *Japan*) being sent to the entity detection. Bert base uncased[6] outperforms Bert large cased (split), and Bert large uncased[7] performs even better for mention detection although not for entity linking (see Table 4).

We also tested the new configuration in combination with Gerbil evaluation, see Table 5. Bert large still performed better than Bert base and using uncased training data resulted in better performances than using cased training data. However the measured absolute differences were larger. For example using the

[5] `https://huggingface.co/Davlan/bert-base-multilingual-cased-ner-hrl`

[6] `https://huggingface.co/dslim/bert-base-NER-uncased`

[7] `https://huggingface.co/Jorgeutd/bert-large-uncased-finetuned-ner`

	NER			InKB Macro F1	InKB Micro F1	Experiment
	Flair					
1	sen	-	cas	78.3%	80.3%	202301160011
	Bert					
8	doc	bas	cas	70.7%	71.0%	202301020003
9	doc	lar	cas	72.9%	72.3%	202301020004
10	spl	bas	cas	72.1%	74.5%	202301020005
11	spl	lar	cas	74.5%	76.3%	202301020006
-	doc	bas	unc	76.1%	77.5%	202301020007
-	doc	lar	unc	78.1%	79.2%	202301020008
12	spl	bas	unc	76.9%	79.4%	202301020009
13	spl	lar	unc	**78.9%**	**81.2%**	202301020010
14	spl	mul	cas	69.2%	74.2%	202301020002

Table 5: Performances measured by Gerbil on entity linking with two different NER software: Flair and Bert, for one Wikipedia version (2019), processing entire documents with a document split value of 500 tokens. The experiment id in the right column links to the page with the full evaluation results at the Gerbil website. The numbers before each row correspond with the row numbers in Table 4. Split documents (spl) performs better than unsplit (doc), uncased (unc) works better than cased (cas) and Bert large (lar) does better than Bert base (bas).

fastest Bert configuration (11: Bert base cased doc) led to a relative drop of 13% in comparison with Flair in Table 4 (F1 for entity linking) and to a 19% loss in Table 5 (InKB Micro F1).

3.5 Error analysis

We compare the output of Bert large split with that of Flair processing sentences. We observed the following behaviour:

Bert performed a lot better than Flair for documents with uninformative case. For example, document 29 from the AIDA/CoNLL-Test B is entirely written in capital characters. Bert identifies all names in *AMERICAN FOOTBALL CONFERENCE EASTERN DIVISION W L T PF PA NEW ENGLAND 9 4 0 355 269 BUFFALO 9 4 0 267 215* correctly but Flair missed three: *American Footbal Conference*, *Eastern Conference* and *New England* (see Figure 5).

Bert was better in identifying names with an initial instead of a first name. For example, in document 17 from the AIDA/CoNLL-Test B data set contains the phrase *S. Chanderpaul c Healy b Blewett 54 C. Hooper run out.* Flair correctly identified *Healy* and *Blewett* as names but only selected *Chanderpaul* and *Hooper*. Meanwhile Bert found all names, including *S. Chanderpaul* and *C. Hooper*.

Bert sometimes unexpectedly picked lower-cased words as names. For example in document 124 from the from the AIDA/CoNLL-Test B data set contains *credited with the most incidents were chow chows, Rottweilers, German shepherds.* Bert only chose *chow chows* as a name (incorrect). Flair selected both

Split value	Server MD F1	InKB EL Micro F1	Experiment id
100	75.0%	78.9%	202301160004
200	75.1%	79.0%	202301160005
300	**75.2%**	79.0%	202301160006
400	74.8%	79.4%	202301160007
500	74.8%	**79.5%**	202301160008
2,000	73.3%	77.5%	202301160009
unsplit	73.3%	77.5%	202301160010

Table 6: Bert base uncased performances on Gerbil by document size split value (maximum processed document length). Longer documents are split to limit text size to the maximum size that Bert can handle (512 tokens). A test with a very large document split value (2,000) was performed to verify that the performance was the same as for unsplit documents. The value of 500 was chosen for the remaining experiments because both the MD and the EL experiments performed well with this threshold.

Rottweilers (debatable) and *German* (correct).

The two systems mostly agree on their selection of names which is reflected in their similar performance in the mention detection task (F=80.3 vs F=81.1, see Table 5). Given the better performance of Bert on uncased text, it is still a surprise to see that their scores are so close. We suspect that Bert's false positives for lower-cased words balances out its advantage on uncased text.

4 Concluding remarks

We replaced the named entity recognition system Flair by Bert in the entity linker REL. We tested different versions of Bert and found that Bert large uncased performed best, slightly better than Flair, while requiring less than half of the processing time that Flair needs (Tables 4 and 5). Because of the document size limitations of Bert, we split long documents in several parts before starting the analysis.

5 References

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[2] Gizem Aydin, Seyed Amin Tabatabaei, Giorgios Tsatsaronis, and Faegheh Hasibi. Find the funding: Entity linking with incomplete funding knowledge bases. In *Proceedings of COLING 2022*. ACL, 2022.

Soccer - Italian First Division Matches This Weekend. Rome 1996-12-06 Italian Serie A games to be played on Sunday (league positions in parentheses , all kick- off times Gmt) : Bologna (4) v Piacenza (13) 1330 Along with leaders Vicenza , fourth-placed Bologna represent the biggest surprise of this Italian autumn. Led as usual by Swede Kennet Andersson and Russian Igor Kolyvanov in attack , Bologna can expect a tough home match against a Piacenza side still exultant after a 3-2 league win over Ac Milan last Sunday. Cagliari (16) v Reggiana (18) 1530 Cagliari start favourite in this relegation scrap following draws with Napoli and Inter in last two outings but will be without suspended Swiss defender Ramon Vega. Bottom team Reggiana are also without a suspended defender , German Dietmar Beiersdorfer. Fiorentina (10) v Perugia (8) 1330 Fiorentina will be without three suspended players – defenders Daniele Carnasciali and Lorenzo Amoruso and midfielder Emiliano Bigica – for a difficult home match against unpredictable , attack-oriented Perugia led by in-form Croat striker Milan Rapajic and the experienced Fausto Pizzi. Lazio (12) v As Roma (7) 1930 Poor man 's Roman derby in what has been a miserable season for both Rome teams , already eliminated from the Italian and Uefa Cups. Lazio have injury doubts about striker Pierluigi Casiraghi , Czech midfielder Pavel Nedved and defender Paolo Negro , while Roma present a full strength side led by Argentine Abel Balbo , Marco Delvecchio and Francesco Totti in attack. Ac Milan (9) v Udinese (11) 1330 Can Milan sink any further ? Following their midweek Champions ' League elimination by Norwegian side Rosenborg , a morale-boosting win is badly needed. Liberian striker George Weah makes a welcome return for Milan alongside Roberto Baggio , with Montenegrin Dejan Savicevic in midfield. Good news for Milan is that Udinese 's German striker Oliver Bierhoff is out through injury. Napoli (5) v Verona (17) 1330 In-form Napoli should prove too strong for second from bottom Verona despite the absence of their suspended Argentine defender Roberto Ayala. Verona 's slim chances have been further reduced by a knee injury to their experienced midfielder Eugenio Corini. Parma (14) v Atlalanta (15) 1330 Parma may field new signing , Croat midfielder Mario Stanic , in an attempt to lift a miserable season which has seen them go without a win since a 1-0 triumph over Cagliari eight weeks ago. Parma 's French midfielder Daniel Bravo and defender Fabio Cannavaro are suspended while Argentine Nestor Sensini is out through injury. Atalanta look to Filippo Inzaghi , scorer of eight goals. Sampdoria (6) v Juventus (3) 1330 All-conquering Juventus field their most recent signing , Portuguese defender Dimas , while Alessandro Del Piero and Croat Alen Boksic lead the attack. The new world club champions may prove too strong for a Sampdoria side led by captain Roberto Mancini but missing injured French midfielder Pierre Laigle. Vicenza (1) v Internazionale (2) 1330 Not exactly a clash of the titans but an intriuguing match nonetheless. Full strength Vicenza , led by Uruguayan Marcelo Otero , may continue their surprise run at the top against an Inter side that has been less than impressive in three successive home draws. Inter will be without suspended French defender Joceyln Angloma and injured Chilean striker Ivan Zamorano

Figure 3: Effect of the Bert text size limit on named entity recognition: all entities beyond a certain point in the text (from *to their*, after 512 tokens) are missed. Correctly identified entities are green while missed and incorrect entities are red.

- mention_detection.find_mentions()
  - syntok tokenizer (when splitting text in sentences)
  - Flair tokenizer (preprocessing for Flair)
  - Bert tokenizer (preprocessing Bert and splitting documents)
  - Nltk tokenizer (when processing context for server.py; replaced by str.split)
- model.predict()
  - uses str.split() and re.split()

Figure 4: Tokenization modules used in REL.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] Hideaki Joko, Faegheh Hasibi, Krisztian Balog, and Arjen P. de Vries. Conversational entity linking: Problem definition and datasets. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. ACM, New York, 2021.

[5] Chris Kamphuis, Faegheh Hasibi, Jimmy Lin, and Arjen P. de Vries. Rebl: Entity linking at scale. In *Proceedings of DESIRES 2022 – 3rd International Conference on Design of Experimental Search & Information REtrieval Systems*, pages 30–31. San Jose, CA, USA, August 2022.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing at NeurIPS 2019*. Neural Information Processing Systems Foundation, 2019.

[7] Anna Tigunova, Paramita Mirza, Andrew Yates, and Gerhard Weikum. PRIDE: Predicting Relationships in Conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4636–4650, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[8] Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Gerbil: general entity annotator benchmarking framework. In *Proceedings of the 24th international conference on World Wide Web*, pages 1133–1143, 2015.

[9] Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference*

NFL AMERICAN FOOTBALL-STANDINGS AFTER THURSDAY'S GAME. NEW YORK 1996-12-05 National Football League standings after Thursday's game (tabulate under won, lost, tied, points for and points against) : AMERICAN FOOTBALL CONFERENCE EASTERN DIVISION W L T PF PA NEW ENGLAND 9 4 0 355 269 BUFFALO 9 4 0 267 215 INDIANAPOLIS 8 6 0 269 284 MIAMI 6 7 0 285 266 NY JETS 1 12 0 221 368 CENTRAL DIVISION W L T PF PA PITTSBURGH 9 4 0 299 211 HOUSTON 7 6 0 291 254 JACKSONVILLE 6 7 0 263 288 CINCINNATI 5 8 0 299 318 BALTIMORE 4 9 0 320 369 WESTERN DIVISION W L T PF PA X-DENVER 12 1 0 351 199 KANSAS CITY 9 4 0 262 230 SAN DIEGO 7 6 0 277 323 OAKLAND 6 7 0 274 234 SEATTLE 5 8 0 250 317 NATIONAL FOOTBALL CONFERENCE EASTERN DIVISION W L T PF PA DALLAS 8 5 0 254 201 WASHINGTON 8 5 0 291 251 PHILADELPHIA 8 6 0 313 302 ARIZONA 6 7 0 248 332 NY GIANTS 5 8 0 200 250 CENTRAL DIVISION W L T PF PA Y-GREEN BAY 10 3 0 346 191 MINNESOTA 7 6 0 243 245 CHICAGO 5 8 0 202 248 DETROIT 5 8 0 263 289 TAMPA BAY 4 9 0 153 243 WESTERN DIVISION W L T PF PA SAN FRANCISCO 10 3 0 325 198 CAROLINA 9 4 0 292 164 ST LOUIS 4 9 0 246 334 ATLANTA 2 11 0 234 393 NEW ORLEANS 2 11 0 184 291 X– CLINCHED DIVISION TITLE Y– CLINCHED PLAYOFF BERTH SUNDAY, DECEMBER 8 ST LOUIS AT CHICAGO BALTIMORE AT CINCINNATI DENVER AT GREEN BAY JACKSONVILLE AT HOUSTON NY GIANTS AT MIAMI ATLANTA AT NEW ORLEANS SAN DIEGO AT PITTSBURGH WASHINGTON AT TAMPA BAY DALLAS AT ARIZONA NY JETS AT NEW ENGLAND BUFFALO AT SEATTLE CAROLINA AT SAN FRANCISCO MINNESOTA AT DETROIT MONDAY, DECEMBER 9 KANSAS CITY AT OAKLAND

NFL AMERICAN FOOTBALL-STANDINGS AFTER THURSDAY'S GAME. NEW YORK 1996-12-05 National Football League standings after Thursday's game (tabulate under won, lost, tied, points for and points against) : AMERICAN FOOTBALL CONFERENCE EASTERN DIVISION W L T PF PA NEW ENGLAND 9 4 0 355 269 BUFFALO 9 4 0 267 215 INDIANAPOLIS 8 6 0 269 284 MIAMI 6 7 0 285 266 NY JETS 1 12 0 221 368 CENTRAL DIVISION W L T PF PA PITTSBURGH 9 4 0 299 211 HOUSTON 7 6 0 291 254 JACKSONVILLE 6 7 0 263 288 CINCINNATI 5 8 0 299 318 BALTIMORE 4 9 0 320 369 WESTERN DIVISION W L T PF PA X-DENVER 12 1 0 351 199 KANSAS CITY 9 4 0 262 230 SAN DIEGO 7 6 0 277 323 OAKLAND 6 7 0 274 234 SEATTLE 5 8 0 250 317 NATIONAL FOOTBALL CONFERENCE EASTERN DIVISION W L T PF PA DALLAS 8 5 0 254 201 WASHINGTON 8 5 0 291 251 PHILADELPHIA 8 6 0 313 302 ARIZONA 6 7 0 248 332 NY GIANTS 5 8 0 200 250 CENTRAL DIVISION W L T PF PA Y-GREEN BAY 10 3 0 346 191 MINNESOTA 7 6 0 243 245 CHICAGO 5 8 0 202 248 DETROIT 5 8 0 263 289 TAMPA BAY 4 9 0 153 243 WESTERN DIVISION W L T PF PA SAN FRANCISCO 10 3 0 325 198 CAROLINA 9 4 0 292 164 ST LOUIS 4 9 0 246 334 ATLANTA 2 11 0 234 393 NEW ORLEANS 2 11 0 184 291 X– CLINCHED DIVISION TITLE Y– CLINCHED PLAYOFF BERTH SUNDAY, DECEMBER 8 ST LOUIS AT CHICAGO BALTIMORE AT CINCINNATI DENVER AT GREEN BAY JACKSONVILLE AT HOUSTON NY GIANTS AT MIAMI ATLANTA AT NEW ORLEANS SAN DIEGO AT PITTSBURGH WASHINGTON AT TAMPA BAY DALLAS AT ARIZONA NY JETS AT NEW ENGLAND BUFFALO AT SEATTLE CAROLINA AT SAN FRANCISCO MINNESOTA AT DETROIT MONDAY, DECEMBER 9 KANSAS CITY AT OAKLAND

Figure 5: Mention detection differences between Flair (top, 38-32) and Bert large split (bottom, 67-4) Green indicates correctly identified entities while red signals entities which were missed or which were classified incorrectly. Bert performs better than Flair on this text because it has also been trained on text without clues from capital letters.

on Research and Development in Information Retrieval (SIGIR '20). ACM, New York, 2020.

Contents

1 Introduction **1**

2 Evaluation of REL **1**
2.1 Evaluating REL with the script `efficiency_test.py` 1
2.2 Evaluating REL with Gerbil 3

3 Optimization **4**
3.1 Replacing Flair with Bert 4
3.2 Evaluating Bert with Gerbil 5
3.3 Error analysis 6
3.4 Applying Bert to documents 6
3.5 Error analysis 9

4 Concluding remarks **10**

5 References **10**