Named Entity Linking to Wikipedia entries: casing Bulgarian

Nikolay Paev, Stefan Marinov, Kiril Simov

CLaDA-BG Conference, Sofia, 25-26 June 2026

Entity recognition and linking

- Named entity recognition - finding a mention of an entity in text
- **Named entity linking** - mapping the mention to an entry in a Knowledge Base

Related works

- Treat entity linking as a ranking problem, not a classification
- Encoding mentions and Entity descriptions in joint vector space
  Gillick et al., CoNLL 2019
- Bi-encoder - Cross-encoder setup for linking to Wikipedia entity
  Wu et al., EMNLP 2020

We employ the Bi-encoder - Cross-encoder setup and make use of the more recent ModernBERT architecture

Data

- Wikipedia description articles - 250K+ for linking, clean

- Training data - 3 datasets:
  - BulTreeBank dataset - 7287 contexts (Org, Per, Loc, …)
  - Balto-slavic dataset - 20060 contexts (Org, Per, Loc, …)
  - Bulgarian Event corpus - 56050 contexts (Org, Per, Loc, **COREF**, …)

- *left context [META] mention [META] right context*
- the context is up to 2 sentences to the left and to the right

Coreference structures in the examples

- Event corpus has coreference structures annotated like NEs
- Coreference is the linguistic relation where two or more things in text refer to the same entity or event

Peter is known for its intelligence. He [Peter] won a math competition when he was a student.

- In the event corpus all mentions (*He*) are annotated with the wikipedia link of their referent.
- Model trained with this data should perform coreference resolution.

System development

- Two-stage approach:
  - Bi-encoder provides candidates
  - Cross-encoder ranks the candidates

- Base model is our own pre-trained ModernBERT model
- 29B Bulgarian tokens
- All our **pre-trained models are available** at

  https://huggingface.co/AIaLT-IICT

BulTreeBank CLaDA BG

First stage

- Model fine-tuned to produce vector representations of texts

- mention with context –Bi-encoder→ vector
- wikipedia article —Bi-encoder→ vector

- The model is trained to produce similar vectors for the positive pairs in the dataset
- After training all wiki articles are vectorized (like indexing)

Second stage

- The Bi-encoder produces context vector. It is compared to the vectors of all wiki articles and initial ranking is achieved

- Cross-encoder is used to rank the top candidates from the Bi-encoder ranking

- Mention with context + Wikipedia article —Cross-encoder→ score

Why two stages

- Vectorizations of the contexts and the wiki articles by the Bi-encoder are **independent**
  - Precompute the wiki article vectors
  - No information sharing when calculating similarities
- Cross-encoder is like classifier
  - Attention between the mention context and the wiki article
  - Cannot be precomputed
- The pipeline approach brings the best of both worlds
  - Bi-encoder produces 32 candidates
  - Cross-encoder scores them independently
- The same approach is widely used in RAG systems

Experiments

- Our datasets provides only positive pairs
  - (mention context, wiki article)

- Bi-encoder model is trained with Multiple Negatives Ranking Loss
  - in-batch negatives
- Only one positive wiki article
  - we consider all other similar wiki articles according to the Bi-encoder as hard negatives - 5 negative pairs for each positive
- Cross-encoder is trained with Cross Entropy Loss
  - positive and (hard) negative pairs

Quantitative results

- Recall@k metric - shows the probability that the correct wiki article is at position ≤ k

- Test set Recall@1:
  - First stage 0.8480
  - Second stage 0.9269

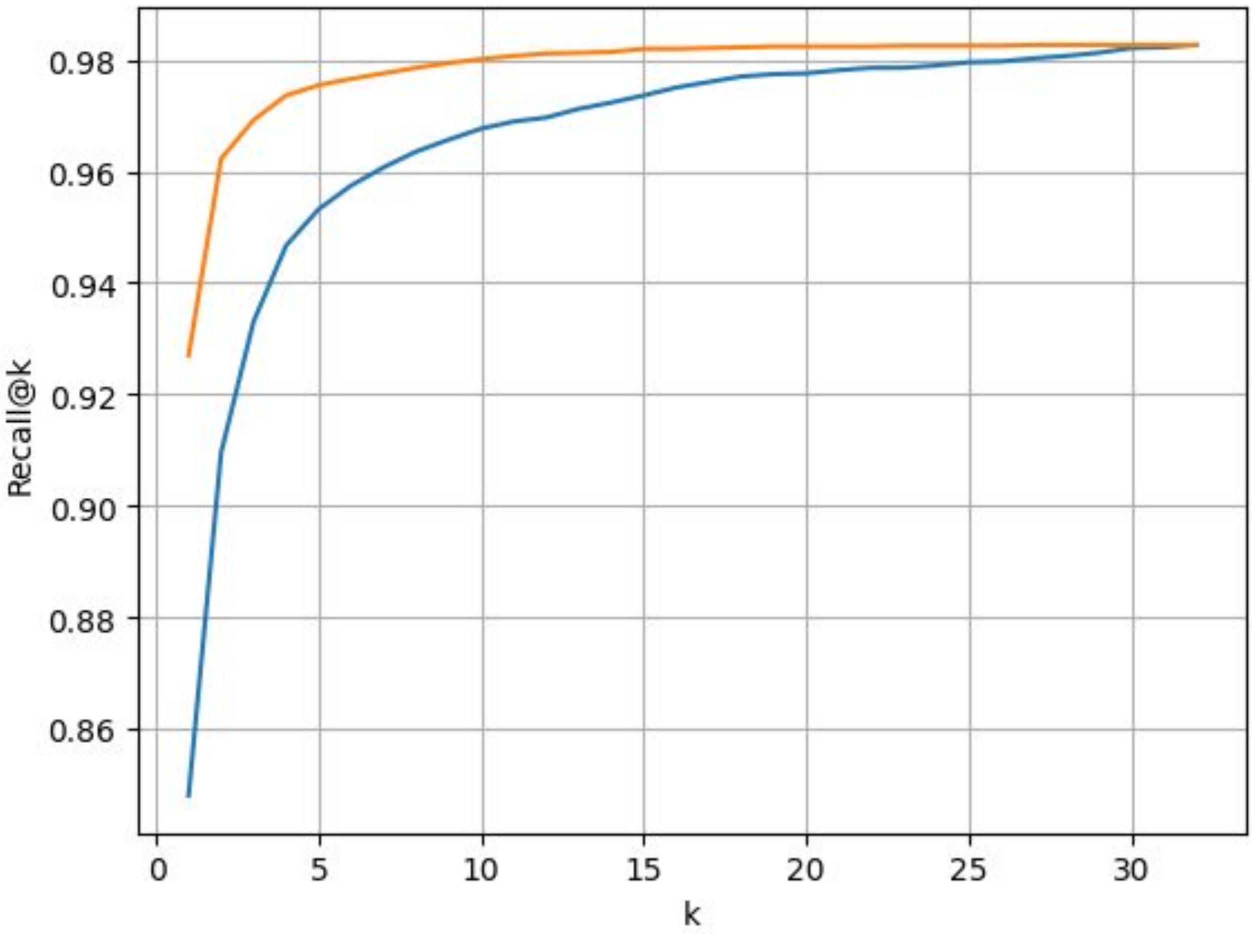
- Test set Recall@3:
  - First stage 0.9330
  - Second stage 0.9692

2nd Stage	PER	LOC	ORG	COREF	EVENT
Recall@1	0.9551	0.9637	0.8931	0.7951	0.9231
Recall@3	0.9778	0.9868	0.9544	0.9061	0.9712

Recall@k

First stage (Bi-encoder)

Second stage (Bi-encoder + Cross-encoder)



Linking example

*Regional court of **Leon** county of the american state of Florida…*

- First stage:
  - 1) Nuevo Leon
  - 2) Leon (Nicaragua)
  - …
  - **22) Leon County, Florida**
- Second stage:
  - **1) Leon County, Florida**
  - 2) Leon (City)
  - …

Manual evaluation

- Manual evaluation on 3 new documents
  - Two articles from “Book for Thessaloniki”
  - One Wikipedia article

- Initial entity recognition provided by NER model

- Entity disambiguation using two-sentence context to the left and two sentence to the right

Successful disambiguation

- Given enough context the homonymous structures are successfully disambiguated

[G. Dimitrov] (George Dimitrov) on the Leipzig Trial (text under photo)

- The correct Wiki article is chosen despite there being 14 articles about a person named George Dimitrov

Coreference resolution

The family Abbott has deep roots and leading positions in [the town] (Thessaloniki) at least from the second half of XVIII century.

Thessaloniki is mentioned only by “the town” but the model correctly predicts the reference.

Difficult cases

The Italian consul Foscarini emphasizes that on the fatal day [Abbott] (Henry Abbott) and [Moulin] (Jules Moulin) went to the konak (residence).

- “Moulin” is mistaken for the town Moulins [Мулин]
- The correct Wiki article is of rank 2

- “Abbott” is correctly disambiguated despite there being 8 more persons with this family name

Conclusion and applications

We developed an automatic Entity disambiguation system which proved efficient and highly accurate.

It is being integrated in the Humanitarian workplace system

- Candidate generation for manual annotation
- Better mention searching after automatic indexing
- Finding entities cooccurrences, getting insights about mentions
- Creating artificial corpora with more dense knowledge
- others

Future work

- The quality of the Wikipedia articles matter significantly
  - Pages where many names are mentioned confuse the system

- Towards creating better entity description vectors combining textual descriptions and averaged mentions in different contexts