JTTE at ATE-IT: A CRF Model with Contextual Embeddings

Juliette Tonneau[1,*], Giorgio Maria Di Nunzio[1]

[1]Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract

Automatic Term Extraction (ATE) is a core task in Natural Language Processing, aiming to identify domain-specific terminology that can support downstream applications such as information retrieval, ontology construction, and domain adaptation. This paper presents a sequence labeling approach to the ATE-IT shared task at EVALITA 2026, focusing on the extraction of terms from Italian institutional texts in the waste management domain. The task is formulated as a BIO tagging problem and addressed using a linear-chain Conditional Random Field (CRF) model. The proposed system integrates symbolic linguistic features obtained from spaCy, contextual semantic representations derived from a pretrained Italian BERT model, and weak domain knowledge through an automatically constructed glossary extracted from the training data. BERT embeddings are used as fixed contextual features and combined with token-level linguistic information within the CRF framework to enforce coherent span-level predictions. Experimental results on both development and test sets show that the proposed approach consistently improves Type F1-score over the baseline, with particularly notable gains in precision, indicating more accurate reconstruction of complete term boundaries. While recall remains lower than the baseline, the analysis highlights the effectiveness of combining contextual embeddings with structured decoding for Italian Automatic Term Extraction and outlines directions for improving generalization and recall in future work.

Keywords

Automatic Term Extraction, Sequence Labeling, Conditional Random Fields, BERT, Italian NLP

1. Introduction

Automatic Term Extraction (ATE) is a fundamental task in Natural Language Processing. Its goal is to identify domain-specific terms within a text [1, 2]. These terms can be multiword or single-word concepts that are relevant to a field. The terms extracted through ATE serve as foundation for multiple tasks such as information retrieval, machine translation, ontology construction, knowledge-graph enrichment, and domain adaptation of large language models (LLM) [3].

The Automatic Term Extraction - Italian Testbed (ATE-IT) [4] shared task is part of EVALITA 2026 [5]. It provides a space to perform ATE in the Italian Language and is centered around the field of waste management in institutional texts. In this domain, we can find single-word terms, synonyms, abbreviations, and multiword terms, which allows us to assess the efficiency of approaches.

The paper presents the solution for Subtask A: Term Extraction. The goal of this subtask is to extract terms relevant to the waste management domain from sentences drawn from a corpus related to the municipal waste management domain.

The remainder of this paper is organized as follows. Section 2 describes the proposed methodology, including problem formulation, linguistic preprocessing, domain glossary construction, feature extraction, and the CRF-based sequence labeling model. Section 3 presents the evaluation setup and experimental results on the development and test sets. Section 4 discusses the strengths and limitations of the system and outlines possible improvements. Finally, Section 5 concludes the paper and summarizes the main contributions of this work.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

julietteannelise.tonneau@studenti.unipd.it (J. Tonneau); giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

0000-0001-7116-9338 (G. M. Di Nunzio)

2. Description of the approach

2.1. Problem Formulation

The task addressed in this work is Automatic Term Extraction (ATE), formulated as a sequence labeling problem. Given a sentence drawn from the waste management domain, the objective is to identify all tokens that belong to domain-specific terms and to reconstruct complete term spans, which may consist of one or multiple consecutive tokens.

Formally, let X be a sequence of T tokens obtained after linguistic preprocessing of a sentence:

$$X = (x_1, ..., x_i, ..., x_T),$$

.

The goal is to predict a corresponding sequence of labels Y

$$Y = (y_1, ..., y_i, ..., y_T),$$

where each $y_i \in \{B - TERM, I - TERM, O\}$. The label B-TERM marks the beginning of a term, I-TERM marks the continuation of a term, and O denotes tokens that are not part of any term. A domain term is defined as any contiguous span of tokens starting with a B-TERM label and optionally followed by one or more I-TERM labels.

The task presents several challenges. First, term boundaries must be identified precisely, as partial matches are not considered correct under span-level evaluation. Second, labels are not independent: valid predictions must respect structural constraints imposed by the BIO scheme. For instance, an I-TERM label cannot occur without a preceding B-TERM. Finally, terms may have significant lexical and syntactic variability, requiring the model to integrate contextual, linguistic, and semantic information.

2.2. Overall Architecture

The system follows a pipeline architecture illustrated in Figure 1.

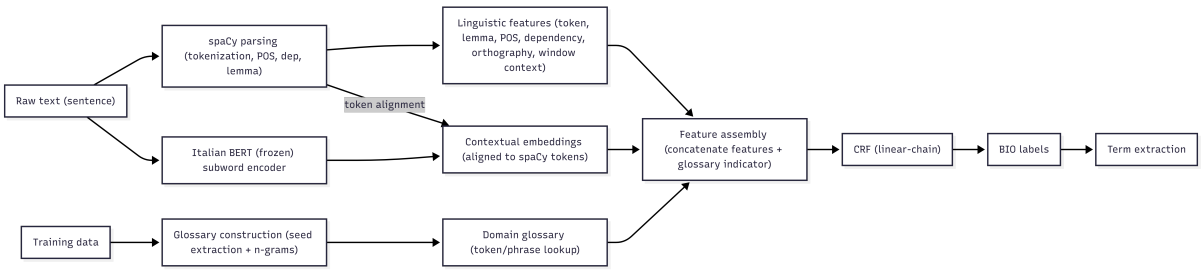

Figure 1: Overall architecture of the system

The main stages are the sentence parsing and linguistic annotations using spaCy, the automatic construction of a glossary from training data, the token-level feature extraction, including BERT-based contextual embeddings, the sequence labeling with a linear-chain Conditional Random Field, and the reconstruction of term spans from predicted BIO labels.

Each part of this pipeline will be described in the following sections.

2.3. Linguistic Preprocessing and Tokenization

All inputs are processed using a linguistic preprocessing pipeline based on the spaCy Italian language model `it_core_news_sm`. This step provides a structured representation of each sentence and establishes the tokenization scheme used consistently throughout training, validation, and inference.

2.3.1. Tokenization

Sentences are first segmented into tokens using spaCy's rule-based tokenizer. Tokenization defines the fundamental units over which features are extracted and labels are predicted. It is necessary to maintain consistent token boundaries, as the next steps operate at the token level.

The spaCy tokenizer is applied uniformly in the `train`, `dev`, and `test` datasets, ensuring alignment between gold annotations, extracted features, and predicted labels. Special attention is given to preserving original token order and indices to support accurate reconstruction of multi-token terms during post-processing.

2.3.2. Linguistic Annotation

For each token, spaCy provides a set of linguistic annotations that are later exploited as symbolic features. These annotations include the following.

- `Lemma`, representing the canonical form of the token
- `Part-of-speech (POS) tags`, marking each word as a syntactic category
- `Morpho-syntactic tags`, adding a precision to the POS tags (eg.: verb tense, noun gender, and number)
- `Dependency relations`, describing the syntactic relations between the tokens
- `Orthographic attributes`, such as the presence of digits
- `Stopword indicators`, allowing the model to distinguish tokens that bear content from functional words

These annotations are stored in a structured token representation and will be used as features for the sequence labeling model.

2.3.3. Contextual Token Information

Beyond individual token attributes, contextual information is extracted by considering neighboring tokens within a fixed window. Concretely, for each token at position i in the spaCy `Doc`, we add the lowercased surface form and the POS tag of the previous token ($i-1$) and of the next token ($i+1$). These features are encoded as `prev_text`, `prev_pos`, `next_text`, and `next_pos`. When the current token is at the beginning or end of the sentence, the missing neighbor is handled using sentinel values: `<START>` for the previous token and `<END>` for the next token.

Features derived from preceding and following tokens encode local syntactic and lexical context, which is particularly important for identifying term boundaries.

2.4. Domain Glossary Construction

To incorporate domain-specific knowledge into the term extraction process, we automatically construct a domain glossary from the training data. The glossary is used as a weak semantic signal during feature extraction, without enforcing hard constraints on predictions.

2.4.1. Domain Seed Extraction

The glossary construction process begins with the identification of domain seed terms from the training corpus. All training sentences are first processed with spaCy to obtain token-level linguistic annotations. Candidate seed tokens are selected if the token is not a stopword or punctuation mark, if the token belongs to a nominal or adjectival part-of-speech category, and if the token is not a temporal expression (e.g., day, names).

Candidate tokens are lemmatized and counted across the corpus. Seeds are selected by retaining the most frequent lemmas with a minimum frequency threshold of 5 occurrences. Among these, only the top 50 most frequent lemmas are kept. This procedure results in 50 domain seed terms, which are stored in an external file (`domain_seeds.txt`) and used as a binary token-level feature during CRF training.

2.4.2. N-Gram Extraction and Filtering

Using the extracted domain seeds, a glossary of candidate multi-word terms is constructed by extracting n-grams (up to a maximum length of six tokens) from the parsed training sentences. N-grams are filtered according to a set of syntactic and orthographic constraints to ensure linguistic validity. In particular, valid n-grams must contain no punctuation, begin and end with nominal or adjectival tokens, and include at least one noun or proper noun.

Additionally, only n-grams containing at least one previously identified domain seed are retained. This constraint ensures that extracted expressions are domain-relevant.

Filtered n-grams are then retained in the final glossary if they occur with sufficient frequency. The resulting glossary thus consists of domain-specific single-word and multi-word expressions that are both linguistically valid and frequent in the data.

The glossary is stored externally and is used during feature extraction to provide a binary indicator that signals whether a given token is present in the glossary.

2.5. Feature Extraction

Feature extraction is performed at the token level and aims to construct a unified representation that integrates symbolic linguistic hints, domain knowledge, and contextual semantic information. Linguistic, contextual window, and domain-related features are derived from spaCy annotations and the automatically constructed glossary, as described in the previous sections.

2.5.1. Contextual Semantic Features with BERT

We enrich each spaCy token with a contextual embedding extracted from a pretrained Italian BERT model (`dbmdz/bert-base-italian-uncased`). Because BERT uses WordPiece subword tokenization whereas the CRF operates on spaCy word-level tokens, we explicitly align subword representations to spaCy tokens.

Given a sentence tokenized by spaCy, we pass the list of spaCy token strings to the HuggingFace tokenizer with `is_split_into_words=True`. This ensures that the tokenizer preserves the mapping between each generated subword and its spaCy token. The sentence is then encoded by the BERT model (with frozen parameters), and we extract contextual representations from the last hidden layer (768 dimensions per subword).

To obtain one vector per spaCy token, we group all subword vectors that share the same word id (`encoded.word_ids()`) and average them (mean pooling). In the rare case where a spaCy token receives no aligned subwords (e.g., due to tokenizer edge cases), we assign a zero vector of the same dimensionality. The resulting 768-dimensional token embedding is concatenated with the symbolic token-level features and provided to the CRF as part of the emission representation. This alignment and mean-pooling strategy is implemented in our `BertEmbedder` module using the HuggingFace `transformers` library.

2.6. Sequence Labeling with Conditional Random Fields

To model dependencies between adjacent labels, the system employs a linear-chain Conditional Random Field (CRF) as the final prediction layer [6]. The model is implemented using the `sklearn_crfsuite` library, a Python interface to CRFsuite[1]. The CRF operates on the token-level feature representations described in the previous section and produces a structured BIO label sequence for each input sentence.

2.6.1. Model Formulation

Let

[1] https://github.com/TeamHG-Memex/sklearn-crfsuite

$$X = (x_1, x_2, ..., x_T)$$

be a sentence of length T, where each x_t is the assembled feature representation of token t, and let

$$Y = (y_1, y_2, ..., y_T)$$

be the corresponding sequence of BIO labels, with $y_t \in \{B - TERM, I - TERM, O\}$.
The CRF defines a conditional probability distribution over label sequences given the input features:

$$p(Y|X) = \frac{1}{Z(X)} exp(\sum_{t=1}^{T}(\psi_{emit}(y_t, x_t) + \psi_{trans}(y_{t-1}, y_t)))$$

[7]
where:

- $\psi_{emit}(y_t, x_t)$ is the emission score, measuring the compatibility between label y_T and the feature vector of token t,
- $\psi_{trans}(y_{t-1}, y_t)$ is the transition score, modeling dependencies between consecutive labels,
- $Z(X)$ is the partition function ensuring normalization.

Emission scores are linear functions of the token-level features, while transition scores are learned parameters capturing valid and frequent label transitions.

2.6.2. Training Objective

The model is trained by maximizing the conditional log-likelihood of the gold label sequences over the training data:

$$\mathcal{L} = \sum_{(X,Y)} \log p(Y|X)$$

Optimization is performed using L-BFGS, with both L1 and L2 regularization applied to diminish overfitting. Hyperparameters controlling regularization strength are selected via randomized search with cross-validation.

2.6.3. Inference

At inference time, the most probable label sequence is obtained by:

$$\hat{Y} = \arg\max_{Y} p(Y|X)$$

which is computed using the Viterbi algorithm. This decoding ensures that predictions respect BIO constraints, such as preventing invalid transitions

2.6.4. Advantages of CRF-Based Decoding

The CRF explicitly models label dependencies and optimizes predictions at the sequence level. This is particularly important for term extraction, where multi-token expressions must be labeled consistently and boundaries must be accurately identified.

By combining rich token-level features and structured decoding, the CRF enforces global coherence in the predicted label sentences.

3. Results and Evaluation

3.1. Evaluation Setup

The system is evaluated on both a development set and a held-out test set using the same preprocessing, feature extraction, and decoding pipeline. Models are trained exclusively on the training data, with hyperparameter selection performed via cross-validation on the training set. The development set is used for validation and qualitative analysis, while the test set is reserved for final performance reporting. This evaluation protocol ensures a clear separation between model selection and final assessment.

The baseline provided by the ATE-IT shared task organizers is a zero-shot large language model approach based on Gemini 2.5 Flash. In this setup, the model is prompted to extract domain-specific terms directly from the input sentences without any task-specific fine-tuning or supervised training. Sentences are processed in batches of 20, and the model outputs are post-processed to obtain the predicted term spans.

3.2. Evaluation Metrics

Performance is evaluated using instance-based and type-based metrics.

Instance-based evaluation (Micro precision, recall, and F1-score) considers each predicted term occurrence as a separate instance. A predicted term instance is counted as correct only if its span exactly matches a gold-standard term instance in the same sentence. Precision is computed as the proportion of correctly predicted term instances among all predicted instances, while recall corresponds to the proportion of gold term instances that are correctly identified. Micro-averaging aggregates true positives, false positives, and false negatives across all sentences before computing the final score.

Type-based evaluation (Type precision, recall, and F1-score) instead operates on the set of unique terms extracted over the corpus. In this setting, multiple occurrences of the same term are counted only once. A predicted term type is considered correct if it exactly matches a gold term type. Precision measures the proportion of correctly predicted unique terms among all predicted unique terms, while recall measures the proportion of gold unique terms that are successfully extracted.

Both evaluations require exact span matching, but they differ in whether frequency of occurrence is taken into account (instance-based) or not (type-based).

3.3. Overall Results

Model	Precision	Recall	F1-score
Micro (Token-level)			
Baseline	0.439	0.616	0.513
Proposed system	**0.717**	**0.674**	**0.695**
Type (Span-level)			
Baseline	0.372	**0.636**	0.470
Proposed system	**0.638**	0.583	**0.609**

Table 1
Performance comparison on the development set between the baseline and the proposed system using token-level (Micro) and span-level (Type) evaluation metrics.

Table 1 reports the performance of the baseline and the proposed system on the development set. The proposed system outperforms the baseline at the token level, achieving higher precision, recall, and F1-score. At the span level, it yields higher precision and F1-score, while the baseline attains higher recall. Overall, the results indicate a clear improvement in extraction accuracy and boundary coherence on the development set.

Table 2 reports the performance of the baseline system and the proposed approach on the test set. At the token level, the baseline achieves the highest recall and F1-score, while the proposed system attains higher precision. At the span level, the proposed system outperforms the baseline in both precision

Model	Precision	Recall	F1-score
Micro (Token-level)			
Baseline	0.497	**0.559**	**0.526**
Proposed system	**0.555**	0.448	0.496
Type (Span-level)			
Baseline	0.435	**0.508**	0.469
Proposed system	**0.561**	0.447	**0.498**

Table 2
Performance comparison on the test set between the baseline and the proposed system using token-level (Micro) and span-level (Type) evaluation metrics.

and F1-score, indicating more accurate reconstruction of complete term spans, whereas the baseline achieves higher recall.

In general, better results are obtained on the development set. This is a common bias in machine learning. Implicitly, the model is adapted to the development set. Indeed, hyperparameters, architectural choices, and feature engineering decisions are selected based on the performance of the model on this set.

4. Discussion

This section analyzes the behavior of the proposed system in the light of the results of the previous section.

4.1. Strengths of the system

A first notable strength of the proposed system is its consistent improvement in span-level (Type) F1-score over the baseline on both the development and test sets. On the development set, the proposed system achieves a higher Type F1-score (0.609 vs. 0.470), and this improvement is maintained on the test set (0.498 vs. 0.469). This indicates that the system is more effective at reconstructing complete term spans with correct boundaries, which is the primary objective of domain-specific term extraction.

At the token level, the system demonstrates higher precision than the baseline on both datasets. On the development set, Micro precision increases from 0.439 to 0.717, while on the test set it increases from 0.497 to 0.555. This suggests that the proposed approach produces fewer false positive labels, reflecting more selective and confident predictions. The combination of contextual BERT embeddings with linguistic features appears to help the model better distinguish term and non-term tokens.

The use of CRF-based structured decoding also contributes positively to output quality. Across both datasets, predictions show coherent BIO label sequences, and extracted terms correspond to contiguous, well-formed spans. This structural consistency is reflected in the improved span-level F1-scores and confirms the benefit of modeling label dependencies explicitly rather than relying on independent token classification.

4.2. Limitations

Despite these improvements, several limitations are apparent from the results.

First, the proposed system exhibits lower recall compared to the baseline at both the token and span levels. On the test set, Micro recall decreases from 0.559 to 0.448 and at the span level, it drops from 0.508 to 0.447. This indicates that while the system is more precise, it tends to miss a portion of relevant terms, especially less noticeable or ambiguous ones.

Second, the difference between development and test performance highlights a degree of generalization difficulty. While gains on the development set are important, improvements on the test set are more moderate, particularly in terms of F1-score. This suggests that some of the learned patterns, such

as those induced by the domain glossary or contextual features, may be sensitive to the distribution of the training and development data.

Finally, although dependency labels and contextual embeddings are used, the model still relies on token-level feature representations. Complex syntactic constructions or long-range dependencies may not be fully captured, which can contribute to boundary errors or missed multi-word expressions.

4.3. Possible improvements

Several paths could be explored to address these limitations.

A first one would be to improve boundary detection by adding different features to the model, such as head information, or explicit noun-phrase boundaries. These features would help the CRF model to decide where spans should start and end.

Improving generalization should be a priority in future improvements. The gap between development and test results suggests a slight overfitting on the data. Improving semantic generalization of BERT features by adding a contrastive embedding regularization and/or using a multi-layer BERT feature mixing could considerably help in this direction.

After those last tasks, another way to upgrade the model would be to implement a two-stage decoding.

In the current system, the CRF performs sequence labeling over all tokens in a sentence, implicitly considering every possible span as a potential term. While this allows for flexible predictions, it also leads to a large search space and encourages conservative behavior, as the model must balance precision and recall across many unlikely span configurations. A two-stage approach addresses this issue by restricting the CRF's decision space to a set of plausible candidate spans. First, the system generates a set of candidate term spans for each sentence. The CRF would then be applied only within the boundaries of candidate spans. By operating within a reduced and more relevant search space, the CRF can make more confident predictions, particularly for multi-token expressions.

5. Conclusion

This paper presented a sequence-labeling approach to Automatic Term Extraction for the ATE-IT shared task at EVALITA 2026, focusing on the waste management domain in Italian. The proposed system combines linguistic preprocessing with contextual semantic representations derived from a pretrained Italian BERT model and leverages a linear-chain Conditional Random Field to enforce structured BIO decoding. Domain knowledge is incorporated through the automatic construction of a glossary extracted from the training data, used as a weak semantic signal during feature extraction.

Experimental results on both development and test sets show that the proposed system consistently improves span-level (Type) F1-score over the baseline, indicating more accurate reconstruction of complete term boundaries. In particular, gains in precision demonstrate the effectiveness of combining contextual embeddings with symbolic linguistic features and structured decoding. However, the analysis also highlights current limitations in generalization and recall.

Overall, the proposed approach provides a strong and extensible baseline for Italian Automatic Term Extraction and offers several promising directions for further improvement within the ATE-IT framework.

Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-5 in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] K. Kageura, B. Umino, Methods of automatic term recognition: A review, Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication 3 (1996) 259–289. doi:10.1075/term.3.2.03kag.

[2] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of Automatic Term Extraction: What happened in 2022?, Digital Scholarship in the Humanities 38 (2023) i41–i47. doi:10.1093/llc/fqad030.

[3] Y. Chun, M. Kim, D. Kim, C. Park, H. Lim, Enhancing Automatic Term Extraction with Large Language Models via Syntactic Retrieval, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Findings of the Association for Computational Linguistics: ACL 2025, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 9916–9926. doi:10.18653/v1/2025.findings-acl.516.

[4] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[5] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[6] N. Nguyen, Y. Guo, Comparisons of sequence labeling algorithms and extensions, in: Proceedings of the 24th International Conference on Machine Learning, ICML '07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 681–688. doi:10.1145/1273496.1273582.

[7] C. Sutton, A. McCallum, An introduction to conditional random fields, 2010. URL: https://arxiv.org/abs/1011.4088. arXiv:1011.4088.