MKTE at ATE-IT: CRF-Based Term Extraction for Italian Waste Management Documents

Minseok Kim[1,*], Giorgio Maria Di Nunzio[1]

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

Abstract

This paper presents a Conditional Random Fields (CRF) sequence-labelling system for sentence-level Automatic Terminology Extraction in the ATE-IT shared task at EVALITA 2026, targeting Italian municipal waste-management documents. The system models term spans using linguistically informed features, including POS tags, lemmas, dependency relations, and morphological patterns, and applies task-specific post-processing to satisfy lowercasing, sentence-level de-duplication, and non-nested output constraints. On the official test set, our submission achieves micro-F1 scores of 0.519 and 0.529, corresponding to ranks 5 and 3 across submitted runs, and yields higher precision than the baseline. However, recall remains comparatively lower, and we observe a substantial gap between training and test performance, suggesting overfitting and limited generalization to unseen documents. We also find only modest gains over a zero-shot large language model approach, indicating that large pre-trained models may encode substantial implicit terminology knowledge even without task-specific supervision. We conclude by outlining directions for improving robustness, including richer semantic representations, data augmentation, and ensemble strategies to reduce the generalization gap.

Keywords

Term Extraction, Conditional Random Fields, Italian NLP, Waste Management

1. Introduction

Automatic Terminology Extraction (ATE) aims to identify and extract domain-specific terms from specialised corpora, with an emphasis on lexical units (single- and multi-word expressions) that denote domain concepts rather than unique real-world referents [1, 2, 3]. Unlike general named entity recognition, ATE focuses on discovering technical terminology that constitutes the conceptual backbone of a domain and is therefore closely connected to downstream tasks such as indexing, knowledge organisation, and domain adaptation.

The ATE-IT task [4] at EVALITA 2026 [5] targets term extraction from Italian municipal waste-management documents, a setting that combines institutional register with domain terminology, abbreviations, and complex multi-word administrative expressions. In Subtask A, systems receive sentences with metadata (e.g., document/paragraph/sentence identifiers) and must extract domain terms (including nouns, verbs, and adjectives) under explicit output constraints: terms must be lowercased, de-duplicated at sentence level, and non-nested. Performance is measured with two complementary metrics: micro-averaged F1 computed over sentence-level instances and type F1 computed over unique term types across the dataset [6].

We model term extraction as a sequence labelling problem using Conditional Random Fields (CRF), which provide a principled framework for predicting structured label sequences while integrating rich, hand-engineered linguistic features [7]. Our system combines token- and context-level cues derived from POS tags, dependency relations, lemmas, and morphology to learn patterns that distinguish domain terms from common words and to improve multi-word boundary detection. We report results on the official test set and submitted runs, and we analyse error profiles to characterise the strengths and limitations of a feature-based CRF approach under ATE-IT's sentence-level constraints.

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

[*] Corresponding author.

minseok.kim@studenti.unipd.it (M. Kim); giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

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

2. Description of the System

Our pipeline consists of four stages: BIO conversion, feature extraction, CRF modeling, and post-processing.

2.1. BIO Tagging and Preprocessing

We convert the training data to sequence labeling format using the BIO (Begin-Inside-Outside) tagging scheme, which is a standard representation for sequence labeling tasks in natural language processing. Each token in a sentence receives one of three labels: B (Begin) marks the first token of a term, I (Inside) marks continuation tokens in multi-word terms, and O (Outside) marks tokens that are not part of any term. This representation allows the model to capture both single-word terms (marked with a single B label) and multi-word terms (marked with B followed by one or more I labels), while also maintaining clear boundaries between adjacent terms.

For each sentence in the training data, we first tokenize the text using SpaCy's Italian tokenizer, which handles Italian-specific linguistic phenomena such as apostrophes, contractions, and punctuation. Then, for each gold-standard term provided in the annotations, we perform case-insensitive substring matching to locate the term within the original sentence text. The matching process operates at the character level to handle potential discrepancies between term boundaries and tokenization boundaries.

When a match is found, we examine which tokens in the SpaCy tokenization overlap with the character span of the term. The first token that overlaps with the term receives the B label, and all subsequent tokens that fall within the term span receive I labels. All tokens that do not overlap with any gold term receive the O label. This character-based approach ensures robust alignment even when the gold term boundaries do not perfectly align with token boundaries, which can occur due to differences in tokenization strategies or the presence of punctuation within terms.

After processing all gold terms for a sentence, we obtain parallel sequences of tokens and BIO labels. These paired sequences constitute the training examples for the CRF model, where each sequence represents a complete sentence context that the model can use to learn patterns of term occurrence and boundary detection.

2.2. Feature Extraction

We extract features using SpaCy's it_core_news_lg model, which provides tokenization, lemmatization, POS tagging, and dependency parsing. For each token, we extract:

Lexical features: lowercase form, boolean indicators for uppercase, title case, and digits, plus 2-character and 3-character prefixes and suffixes helping capture patterns.

Morpho-syntactic features: POS tag indicating grammatical category and dependency relation showing syntactic function, which help distinguish technical terms from functional words.

Lexical-semantic features: lemma providing the dictionary form for generalization, and stop word indicator identifying common functional words.

Orthographic features: token shape abstracting the pattern (e.g., "Xxxxx" for title case, "dddd" for digits).

Contextual features: lowercase form, POS tag, and lemma from the previous and next tokens, enabling the model to learn sequential patterns. Sentence boundary tokens receive special BOS/EOS markers.

Bias feature: constant value allowing the model to learn label priors.

This feature set balances information with efficiency, providing linguistic context, while maintaining reasonable dimension.

2.3. CRF Model

We employ a linear-chain Conditional Random Field that predicts label sequences conditioned on inputs. Unlike local classifiers, CRFs model dependencies between adjacent labels, ensuring coherent BIO sequences. We use sklearn-crfsuite with L-BFGS optimization, L1 and L2 regularization coefficients of 0.1, maximum 200 iterations, and all possible transitions enabled (to learn state transition constraints).

For training, we combine the provided training and development datasets. The CRF learns feature weights maximizing the conditional probability of correct label sequences. We validated our approach using 10-fold cross-validation on the combined data.

2.4. Prediction and Post-processing

For prediction on new sentences, we apply the same feature extraction pipeline used during training. Each sentence is tokenized using SpaCy, and for each token, we extract the same comprehensive feature dictionary described in Section 2.2. This produces a sequence of feature dictionaries representing the sentence.

The trained CRF model then predicts the most likely BIO label sequence for the sentence using the Viterbi algorithm, which efficiently computes the highest-probability path through the label sequence space by using dynamic programming. The Viterbi algorithm takes into account both the emission probabilities (how likely each label is for each token given its features) and the transition probabilities (how likely each label is to follow the previous label), finding the globally optimal label sequence rather than making independent per-token decisions.

After obtaining the predicted BIO label sequence, we reconstruct the extracted terms through a simple scanning process. We iterate through the tokens and their predicted labels from left to right. When we encounter a B label, we start a new term and add the corresponding token text to it. Any following I labels extend the current term by appending their token text (with spaces between tokens). When we encounter an O label or reach the end of the sentence, we finalize the current term and add it to the list of extracted terms for that sentence.

Finally, we apply post-processing steps to ensure strict compliance with the task requirements specified in the ATE-IT guidelines. All extracted terms are converted to lowercase, as required by the task specification. This normalization step ensures consistency regardless of the original capitalization in the source text. Within each sentence, we remove duplicate terms, keeping only unique term strings. If the same term is extracted multiple times from the same sentence (which can occur if it appears in multiple locations), only one instance is retained in the output. The extracted terms are formatted as a JSON structure with the required fields: document_id, paragraph_id, sentence_id (copied from the input), and term_list (containing the list of unique, lowercased terms extracted from that sentence).

This post-processing stage ensures that our system output exactly matches the format expected by the evaluation system and satisfies all task constraints, including the prohibition against nested terms (which is implicitly handled by the BIO scheme, as each token can belong to at most one term).

2.5. Implementation

The complete system is implemented in Python 3. The main dependencies are: SpaCy 3.7.2 with the it_core_news_lg-3.7.2 model for Italian natural language processing, providing tokenization, lemmatization, POS tagging, dependency parsing, and morphological analysis. sklearn-crfsuite 0.3.6 for CRF modeling. scikit-learn 1.5.1 for cross-validation infrastructure and evaluation utilities. pandas 2.2.2 for structured data manipulation and result aggregation. tqdm 4.66.2 for progress bars during data processing and model training.

The system architecture follows a modular design with clear separation of concerns. Separate Python modules handle data loading and parsing, BIO conversion, feature extraction, model training, prediction, and evaluation. This modularity facilitates experimentation with different feature sets, model configurations, and post-processing strategies. The codebase is organized to make it straightforward to modify individual components (such as adding new features or changing the CRF hyperparameters) without affecting other parts of the pipeline.

Table 1
Official test set results

System	Rank	Micro-P	Micro-R	Micro-F1
Baseline (Gemini)	–	0.497	0.559	0.526
Our CRF (Run on 1)	5	0.569	0.476	0.519
Our CRF (Run on 2)	3	0.654	0.444	0.529

Table 2
10-fold cross-validation on training data

Metric	Mean	Std Dev
Micro-Precision	0.799	0.023
Micro-Recall	0.698	0.031
Micro-F1	0.745	0.019
Type-Precision	0.761	0.028
Type-Recall	0.628	0.034
Type-F1	0.687	0.022

3. Results

We submitted a run with minor variations in configuration. Table 1 shows official test results compared to the Gemini-based baseline.

Run on 2 achieved micro-F1 of 0.529 (rank 3), slightly outperforming the baseline (0.526). Run on 1 obtained 0.519 (rank 5), performing comparably to baseline. Both runs show substantially higher precision than baseline (0.569 and 0.654 vs. 0.497) but lower recall (0.476 and 0.444 vs. 0.559). Run 2's higher precision (0.654) comes at the cost of lower recall (0.444), indicating more conservative predictions.

Table 2 shows cross-validation results on training data.

Cross-validation results (micro-F1: 0.745) significantly exceed test performance (0.519-0.529), that probably indicates overfitting to training patterns or systematic differences between training and test distributions.

4. Discussion

4.1. Performance Analysis

Our results reveal several key patterns. Both runs favor precision over recall, producing more conservative predictions than baseline. This suggests our feature-based CRF learned to be selective, identifying high-confidence terms while missing borderline cases. The baseline's higher recall may indicate the will to extract more candidates, though with more false positives.

The large gap between cross-validation (0.745) and test performance (0.519-0.529) is concerning. Several factors may contribute: the test set may contain vocabulary or terminology patterns underrepresented in training data. Our features may have captured training correlations that don't generalize. The training corpus size may limit learning really robust patterns. Municipal waste management terminology varies across regions and administrative contexts, and our model may lack exposure to more test-set terminology.

Our marginal improvement over the zero-shot Gemini baseline, despite using supervised learning with annotated data, suggests large language models possess substantial implicit knowledge about technical terminology. However, the baseline's lower precision indicates it extracts too many spurious terms, while our CRF's higher precision shows it learned more discriminative patterns.

4.2. Error Analysis and Limitations

Several factors likely limit recall performance:

Rare terms: Many domain terms appear infrequently in training data. CRFs require multiple examples to learn reliable patterns. Rare terms, especially those with unusual structures may not be extracted consistently.

Multi-word complexity: Complex noun phrases like ”impianto di trattamento rifiuti” require learning both syntactic patterns and domain relevance. If similar structures appear with non-term content in training, generalization will probably fail.

Domain coverage: Waste management includes multiple sub-domains: regulatory language, waste categories, collection procedures, and infrastructure. Unbalanced training coverage probably leads to uneven performance across the content types.

Context-dependence: Some words are domain terms only in specific contexts. For example, ”conferire” (deliver) becomes technical in ”conferire rifiuti” (deliver waste). Training examples may not clearly demonstrate such distinctions sometimes.

Feature limitations: Our features capture morpho-syntactic patterns but lack semantic representations. Word embeddings or contextual representations from pre-trained language models might provide richer semantic information.

Nested terms: The task prohibits nested terms, yet related terms may overlap. The BIO scheme must choose one representation, and our model may not resolve such cases consistently with gold annotations.

4.3. What Worked

Despite limitations, several aspects proved effective. Rich linguistic features enabled learning discriminative patterns, with POS tags particularly helpful in constraining predictions to appropriate word classes. The CRF’s structured prediction ensures coherent BIO sequences, reducing invalid transitions and improving boundary detection. High precision means extracted terms are generally correct, valuable for applications where false positives are costly.

4.4. Future Improvements

Several directions could improve performance:

Enhanced semantics: Incorporating embeddings or contextualized representations from Italian language models (BERT, RoBERTa) could provide richer semantic information, helping with rare terms and context-dependent terminology.

Ensemble methods: Combining our CRF with neural models, rule-based extractors, or LLM outputs could leverage complementary strengths, balancing precision and recall.

Domain adaptation: Fine-tuning pre-trained models on unlabeled waste management texts or incorporating external resources like glossaries could provide additional supervision.

5. Conclusion

We presented a CRF-based approach to term extraction for Italian waste management documents, achieving competitive performance with micro-F1 scores of 0.519 and 0.529 (ranks 5 and 3). While demonstrating higher precision than baseline, lower recall and substantial train-test performance gaps highlight probable overfitting and generalization challenges. The modest improvement over zero-shot language models suggests that large pre-trained models possess considerable implicit terminology knowledge. Future work should focus on enhanced semantic representations, data augmentation, and ensemble methods to bridge the generalization gap.

Declaration on Generative AI

The author have employed Generative AI tools in the preparation of this work. All code development, experimentation, analysis, and writing were performed by the author with assistance from large language models or other AI generation systems. The author also used Claude 4.5 and ChatGPT to check grammar and spelling. 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] K. Frantzi, S. Ananiadou, H. Mima, Automatic recognition of multi-word terms:. the C-value/NC-value method, International Journal on Digital Libraries 3 (2000) 115–130. doi:10.1007/s007999900023.

[3] 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.

[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] R. Verborgh, M. Vander Sande, P. Colpaert, S. Coppens, E. Mannens, R. Van de Walle, Web-scale querying through linked open data, Semantic Web 9 (2018) 49–69.

[7] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the 18th International Conference on Machine Learning (ICML), 2001, pp. 282–289.