# Binary Token-Level Classification with DeBERTa for All-Type MWE Identification: A Lightweight Approach with Linguistic Enhancement

**Anonymous ACL submission**

## Abstract

We present a comprehensive approach for multiword expression (MWE) identification that combines binary token-level classification, linguistic feature integration, and data augmentation. Our DeBERTa-v3-large model achieves 69.8% F1 on the CoAM dataset, surpassing the best results (Qwen-72B, 57.8% F1) on this dataset by 12 points while using 165× fewer parameters. We achieve this performance by **(1)** reformulating detection as binary token-level START/END/INSIDE classification rather than span-based prediction, **(2)** incorporating NP chunking and dependency features that help discontinuous and NOUN-type MWEs identification, and **(3)** applying oversampling that addresses severe class imbalance in the training data. These results demonstrate that carefully designed smaller models can substantially outperform LLMs on structured NLP tasks, with important implications for resource-constrained deployments.

## 1 Introduction

Multiword expressions (MWEs) challenge NLP systems due to their syntactic flexibility and semantic non-compositionality (Constant et al., 2017; Sag et al., 2002). They are also widely acknowledged to be frequent in natural language (Baldwin and Kim, 2010). Accurate detection nevertheless remains difficult, particularly for discontinuous patterns in which components may be separated by intervening material, as in *look the information up*, where *look...up* forms the MWE while *the information* does not (Rohanian et al., 2019).

MWE identification is traditionally formulated as sequence-labeling tasks using BIO (Ramshaw and Marcus, 1995) or extended tagsets (e.g., BILOU (Ratinov and Roth, 2009)) to capture discontinuities. These multi-class labeling schemes assign each token a single label from a predefined tagset. Current state-of-the-art approaches often focus on specific MWE types, e.g., transformer-based models (Swaminathan and Cook, 2023; Tanner and Hoffman, 2023) that incorporate linguistic features like dependency parsing for verbal discontinuous patterns (Nasr et al., 2015) and chunk features for nominal detection (Cordeiro et al., 2016). Existing methods for all-type MWE rely on massive language models with billions of parameters (Ide et al., 2025) and lack systematic integration of linguistic knowledge. Moreover, while binary classification approaches have proven successful for span-based NER (Eberts and Ulges, 2020; Li et al., 2023; Tan et al., 2020; Fei et al., 2021) to the best of our knowledge this has not been explored yet for the task of MWE identification.

Several datasets support MWE research. PARSEME shared tasks (Savary et al., 2017; Ramisch et al., 2018, 2020; Walsh et al., 2018) provide multilingual corpora for verbal MWEs. DiMSUM (Schneider et al., 2016) covers multiple types but suffers from annotation inconsistencies (Tanner and Hoffman, 2023). STREUSLE (Schneider et al., 2018) augments the English Web Treebank with various MWE types but emphasizes supersense labeling over explicit MWE type distinctions. The recently introduced CoAM (Ide et al., 2025) addresses these limitations by offering comprehensive coverage of all MWE types with explicit type annotations (NOUN, VERB, MOD/CONN, CLAUSE, OTHER), enabling fine-grained analyses previously unavailable. While limited to 1,301 sentences, CoAM's careful curation provides crucial evaluation capabilities.

Leveraging CoAM's comprehensive annotations, we present an approach that reformulates MWE identification as binary token-level classification—departing from traditional multi-class sequence labeling to predict three independent binary labels per token. Our DeBERTa-v3-large model (He et al., 2021) achieves 69.8% F1 on CoAM, surpassing the state-of-the-art on this dataset, Qwen-72B (57.8% F1), by 12 points while using 165×

fewer parameters.

## 2 Method

We detail our approach, beginning with the dataset description and our binary token-level classification framework, followed by linguistic feature integration and data augmentation.

### 2.1 Dataset

CoAM (Ide et al., 2025) contains 1,301 English sentences (780 train, 521 test) with 867 annotated MWEs, distributed over the following types: NOUN (34.3% compounds and idiomatic nominals), VERB (37.5% phrasal and light verb constructions), MOD/CONN (23.9% complex modifiers, CLAUSE (1.6% sentential idioms), and OTHER (2.8%). MWEs are 88.7% continuous and 11.3% discontinuous, spanning 2-13 tokens.

### 2.2 Binary Token-Level Classification and MWE Reconstruction

We reformulate MWE identification as binary token-level classification, inspired by its success in NER (Eberts and Ulges, 2020; Li et al., 2023; Tan et al., 2020; Fei et al., 2021): instead of assigning each token a single label from a multi-class tagset (e.g., B-MWE, I-MWE, O), we make three independent binary decisions per token, predicting whether it serves as a START, END, or INSIDE component of a MWE. Our approach predicts three independent probabilities for each token $x_i$: $p_i^{start}$, $p_i^{end}$, and $p_i^{inside}$. For instance, in *looked the information up*:

- *looked*: START=1, END=0, INSIDE=0
- *the*: START=0, END=0, INSIDE=0
- *information*: START=0, END=0, INSIDE=0
- *up*: START=0, END=1, INSIDE=0

Our model would ideally predict high START probability for *looked*, high END probability for *up*, and low INSIDE probabilities for the intervening words, correctly identifying the discontinuous MWE {*looked*, *up*}. Crucially, our binary token-level approach scales linearly with sequence length $O(n)$, as we make three independent predictions per token, whereas span-based enumeration requires quadratic complexity $O(n^2)$ to evaluate all possible token pairs.

To train this binary classification framework, we must first convert CoAM's original span-based annotations. CoAM provides span-based annotations (token indices for each MWE), which we convert to token-level labels through a projection system generating three binary vectors that form our START/END/INSIDE encoding scheme: $start[i] = 1$ if token $i$ is the first token of any MWE, $end[i] = 1$ if token $i$ is the last token of any MWE, and $inside[i] = 1$ if token $i$ appears strictly between start/end positions. Projections are stored as versioned JSON artifacts with SHA256 hashing for reproducibility.

At inference, these independent predictions must be reconstructed into MWE spans. We form candidate spans from token pairs $(s, e)$ with $e > s$ such that $p_s^{start} \geq \tau_{start}$ and $p_e^{end} \geq \tau_{end}$. A span is accepted if its width $(e - s + 1) \leq 13$ (dataset maximum), intermediate tokens $t \in (s, e)$ are included when $p_t^{inside} \geq \tau_{inside}$, and the reconstructed expression contains 2–6 members, matching CoAM's MWE length distribution. Candidates are stored as sorted sets of token indices. After training, we tune the three thresholds ($\tau_{start}$, $\tau_{end}$, $\tau_{inside}$) independently on the development set via grid search, finding optimal values that maximize the F1 score.

### 2.3 Linguistic Feature Integration

We enrich DeBERTa-v3-large with two forms of syntactic knowledge:

**NP Chunk Features.** Tokens are tagged as inside or outside a noun phrase (NP) using spaCy (Honnibal et al., 2020). These binary tags are mapped to 16-dimensional learnable embeddings and concatenated with contextualized token states. Chunk embeddings help the model detect nominal MWEs such as *stock market*, a class noted in the CoAM study as particularly difficult to recall compared to MOD/CONN or VERB MWEs.

**Dependency features.** Following extensive prior work on dependency-based MWE identification (Lin, 1999; Wehrli, 2000; Green et al., 2013; Nasr et al., 2015), we compute shortest dependency path lengths between all token pairs using NetworkX (Hagberg et al., 2008) graph traversal, capped at 5. These distances serve dual purposes: (1) as learnable embeddings during encoding, where we average each token's dependency distances to other tokens and map them to 32-dimensional embeddings, and (2) as hard constraints during reconstruction, where we reject discontinuous MWE candidates with consecutive members having dependency distance greater than 4, filtering out implausible token combinations.

| | Q-72B | BBS | BBT | BLT | DBT | DLT | DBT+l | DLT+l | DBT+o | DLT+o | DBT+a | DLT+a | DBT+lo | **DLT+lo** | DBT+la | DL+la |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Prec** | 63.8 | 51.6 | 62.5 | 67.4 | 62.1 | 64.0 | 64.0 | 52.6 | 65.0 | 61.3 | 68.2 | 61.1 | 63.6 | **69.3** | 66.4 | 70.0 |
| **Rec** | 52.8 | 55.9 | 55.6 | 53.3 | 65.4 | 61.2 | 68.0 | 69.3 | 61.9 | 63.5 | 58.5 | 69.3 | 66.4 | **70.3** | 63.8 | 66.4 |
| **F1** | 57.8 | 53.7 | 58.9 | 59.5 | 63.7 | 62.6 | 65.9 | 59.8 | 63.4 | 62.4 | 63.0 | 64.9 | 65.0 | **69.8** | 65.1 | 68.2 |

Table 1: Performance comparison of our 15 model configurations on CoAM test set. Our DeBERTa-v3-large using binary token-level classification with linguistic features and oversampling (DLT+lo) achieves 69.8% F1, outperforming the Qwen-72B baseline (57.8% F1).

| | Q-72B | BBS | BBT | BLL | DBT | DLT | DBT+l | DLT+l | DBT+o | DLT+o | DBT+a | DLT+a | DBT+lo | **DLT+lo** | DBT+la | DLT+la |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CL** | 28.6 | 42.9 | 28.6 | 42.9 | 42.9 | 42.9 | 42.9 | 42.9 | 42.9 | 42.9 | 28.6 | 42.9 | 28.6 | **85.7** | 42.9 | 71.4 |
| **M/C** | 57.7 | 67.6 | 75.7 | 76.6 | 82.9 | 76.6 | 80.2 | 82.0 | 76.6 | 79.3 | 67.6 | 78.4 | 76.6 | **84.7** | 73.0 | 80.2 |
| **NOUN** | 42.1 | 43.8 | 38.0 | 35.5 | 57.9 | 51.2 | 60.3 | 65.3 | 53.7 | 52.9 | 53.7 | 62.7 | 60.3 | **66.9** | 63.6 | 58.7 |
| **VERB** | 59.7 | 59.0 | 56.8 | 51.1 | 59.7 | 59.0 | **66.9** | 64.7 | 59.0 | 62.6 | 58.3 | 66.2 | **66.9** | **66.9** | 59.0 | 63.3 |

Table 2: Type-specific recall (%) across MWE types. CL=CLAUSE (N=7), M/C=MOD/CONN (N=111), NOUN (N=121), VERB (N=139). OTHER category (N=3) excluded as no model achieved meaningful recall.

### 2.4 Oversampling

Beyond linguistic features, addressing the severe class imbalance in CoAM's 780 training sentences proved crucial for model performance. We evaluated two data augmentation approaches to address class imbalance (He and Garcia, 2009). First, we tested augmentation via lexical substitution, replacing single tokens outside MWE boundaries with semantically similar words. Second, we explored simple duplication of training sentences. Both approaches were tested by selecting 10%, 20%, 30%, or 40% of the training sentences for augmentation.

Contrary to expectations, duplication outperformed lexical substitution on all metrics, with 30% selection ratio proving optimal. We attribute this to CoAM's limited size combined with its carefully curated annotations—the dataset appears too small for the model to learn invariances from lexical variations, while any modifications risk disrupting the subtle patterns that define MWEs.

## 3 Experiments

We conduct a systematic ablation study with 15 model configurations to isolate the contributions of each component: our binary token-level classification approach, model architecture choice, linguistic feature integration, and data augmentation.

### 3.1 Experimental Design

Our experiments follow four phases. First, we validate our approach by comparing BERT-base span-based prediction (BBS) against BERT-base/large binary token-level classification (BBT/BLT). Second, we test DeBERTa-v3-base/large variants (DBT, DLT), whose disentangled attention—separately encoding content and position—should better capture discontinuous patterns (He et al., 2021). Third, we augment these DeBERTa-v3 models with linguistic features (DBT+l, DLT+l) targeting NOUN and discontinuous MWEs. Finally, we test oversampling and lexical substitution on DeBERTa-v3 (DBT+o, DLT+o, DBT+a, DLT+a, DBT+lo, DLT+lo, DBT+la, DLT+la) both independently and combined with linguistic features.

### 3.2 Setup

We convert CoAM's span-based annotations to our START/END/INSIDE format. We train all models on NVIDIA GH200 hardware. Our baseline is the CoAM authors' Qwen-72B (57.8% F1, (Ide et al., 2025)), with BERT-base span-based as additional comparison. We report exact span matching micro F1 with breakdowns by type and continuity. Our best model (DLT+lo) uses learning rate 3e-5, batch size 16, dropout 0.3, weight decay 0.01 with early stopping triggered at epoch 8. Training requires 10-25 minutes per model, making our approach highly practical compared to large model fine-tuning. We focus on CoAM as it provides comprehensive type annotations unavailable in PARSEME's verb-only focus or DiMSUM's inconsistent annotations.

### 3.3 Main Results

Our systematic ablation results in Table 1 reveal several critical insights about each component's contribution. Our binary token-level classification yields +5.2 F1 improvement over span-based prediction (BBS: 53.7 → BBT: 58.9), validating

that three independent START/END/INSIDE decisions per token provide richer gradient signals than sparse span-level feedback. DBT achieves 63.7% F1, surpassing BBT by +4.8 points and even outperforming BLT, confirming that disentangled attention mechanisms are particularly suited for MWE detection where content and position interact complexly.

Adding linguistic features to DBT improves F1 by +2.2 points (63.7 → 65.9), with particularly strong recall improvements. However, DLT shows a surprising decrease with linguistic features alone (62.6 → 59.8), suggesting potential overfitting on the small dataset. A critical pattern emerges in the interaction between augmentation and model capacity: without augmentation, base models often outperform large counterparts, but with augmentation, large models excel—DLT+lo achieves 69.8% F1 while DBT+lo reaches only 65.0%. Oversampling consistently outperforms lexical substitution when combined with linguistic features, which we attribute to CoAM's high annotation quality where lexical changes risk disrupting subtle MWE patterns (see Appendix A.2 and Appendix B for detailed discussion).

Our DeBERTa-v3-large model achieves 69.8% F1 on CoAM, surpassing the state-of-the-art on this dataset, Qwen-72B (57.8% F1), by 12 points while using 165× fewer parameters. This approach differs fundamentally from recent frameworks like Swaminathan and Cook (2023) that propose a method that is tailored specifically to verbal MWEs or binary idiomaticity classification rather than comprehensive span identification across all MWE types, which would introduce biases when applied directly to the CoAM dataset (see Framework Comparison Limitations).

### 3.4 Analysis by MWE Type and Continuity

Table 3[1] highlights dramatic improvements for discontinuous MWEs. The baseline span-based model completely fails on discontinuous MWEs (0% recall), while our binary token-level classification immediately enables detection (BBT: 23.3% recall). Adding linguistic features to DBT increases discontinuous recall by +4.6 points, confirming that explicit syntactic knowledge helps identify separated MWE components. Our best model (DLT+lo) achieves 34.9% discontinuous recall while maintaining 25.9% precision (see Appendix B for detailed error analysis).

[1]Type-specific precision cannot be computed as our model predicts only reconstructed span boundaries without type labels. See Limitations (Type Prediction Limitations) for detailed discussion.

| | C-P | C-R | C-F1 | D-P | D-R | D-F1 |
|---|---|---|---|---|---|---|
| Q-72B | – | – | 57.3 | – | – | 17.1 |
| BBS | 51.6 | 63.0 | 56.6 | 0.0 | 0.0 | 0.0 |
| BBT | 77.4 | 59.8 | 67.4 | 12.8 | 23.3 | 16.5 |
| BLT | **81.1** | 58.3 | 67.8 | 10.3 | 14.0 | 11.9 |
| DBT | 77.5 | 70.4 | 73.8 | 11.7 | 25.6 | 16.1 |
| DLT | 78.9 | 66.3 | 72.0 | 11.3 | 20.9 | 14.6 |
| DBT+l | 74.1 | 72.8 | 73.4 | 17.8 | 30.2 | 22.4 |
| DLT+l | 65.7 | **74.9** | 70.0 | 9.4 | 25.6 | 13.8 |
| DBT+o | 79.1 | 66.0 | 71.9 | 16.0 | 30.2 | 21.0 |
| DLT+o | 76.5 | 67.5 | 71.7 | 14.4 | 32.6 | 20.0 |
| DBT+a | 80.2 | 62.1 | 70.0 | 20.0 | 30.2 | 24.1 |
| DLT+a | 76.5 | 73.1 | 74.7 | 15.6 | **39.5** | 22.4 |
| DBT+lo | 72.5 | 70.1 | 71.3 | 22.5 | 37.2 | 28.1 |
| **DLT+lo** | 76.9 | **74.9** | **75.9** | **25.9** | 34.9 | **29.7** |
| DBT+la | 75.8 | 67.8 | 71.6 | 21.9 | 32.6 | 26.2 |
| DLT+la | 80.4 | 70.4 | 75.1 | 23.1 | 34.9 | 27.8 |

Table 3: Performance on continuous (C) and discontinuous (D) MWEs. P=Precision, R=Recall, N=338 continuous, N=43 discontinuous. DLT+lo achieves best overall balance with 75.9% continuous F1 and 29.7% discontinuous F1.

In addition to continuity patterns, we also analyze performance across MWE types to understand where improvements are most pronounced. Table 2 demonstrates type-specific improvements across categories. CLAUSE expressions improve from Qwen-72B's 28.6% to 85.7% recall, though with only 7 test instances, this represents detecting 6 versus 2 examples. NP chunking knowledge integration helps NOUN expressions, improving recall from 57.9% to 60.3% (DBT → DBT+l), with stronger effects in large models. MOD/CONN expressions show consistent gains, with DLT+lo achieving 84.7% recall compared to Qwen's 57.7%.

## 4 Conclusion

We reformulate MWE identification as binary token-level classification with three independent START/END/INSIDE predictions per token, combining DeBERTa-v3-large with NP chunking and dependency features plus 30% oversampling. This approach achieves 69.8% F1 on CoAM, outperforming Qwen-72B (57.8% F1) by 12 points with 165× fewer parameters. For discontinuous MWEs, we achieve 29.7% F1 versus Qwen-72B's 17.1%, doubling baseline performance though these patterns remain challenging.

## Limitations

While our approach achieves substantial improvements on CoAM, several limitations warrant discussion:

**Dataset Scale and Model Generalization.** Our evaluation focuses on the CoAM dataset with 780 training and 521 test sentences. Although this limited scale leads to potential overfitting patterns (see Appendix A.2 for detailed analysis), meaningful generalization is evident: the test set remains unseen during training and is guaranteed to not contain MWE patterns present in the training data, yet our model maintains strong performance. The performance difference between training and test sets (Table 5) is typical of small datasets. The finding that simple duplication outperforms lexical substitution in our oversampling experiments (Section 2.4) suggests that preserving exact MWE patterns is crucial for this task, as even minor lexical variations can fundamentally alter multiword expressions' idiomatic properties. Future work should evaluate on additional datasets like DiMSUM to further assess cross-dataset generalization, though CoAM's comprehensive type annotations and consistency-checked quality provide unique evaluation capabilities unavailable elsewhere.

**Statistical Robustness.** Preliminary experiments with different random seeds on a subset of models (5 out of 15 configurations) showed no significant performance variation from the reported results using seed 42, suggesting stability in our approach. However, due to limited time for resource access, we could not systematically test all model configurations across multiple seeds.

**Framework Comparison Limitations.** Recent work by Swaminathan and Cook (2023) achieves strong results on MWE-related tasks but addresses fundamentally different problems than comprehensive span detection, preventing direct comparison. Their MTLB-STRUCT framework targets PARSEME 1.2, which exclusively annotates verbal MWEs (light-verb constructions, verb-particle constructions, inherently reflexive verbs, and verbal idioms), making it architecturally incompatible with CoAM where 62.5% of MWEs are non-verbal (NOUN, MOD/CONN, CLAUSE, OTHER categories). Their second approach addresses SemEval 2022 Task 2, a binary sentence-level classification task determining whether a known potentially-idiomatic expression is used idiomatically or literally—fundamentally different from identifying which token spans constitute MWEs.

**Discontinuous MWE Performance.** Despite doubling the baseline's recall for discontinuous MWE identification (34.9% vs 17.1%), the overall discontinuous F1 remains modest at 29.7%. The dependency distance constraints we introduce (rejecting candidates with consecutive members having distance >4) improve precision but may be overly restrictive for certain discontinuous patterns. This fundamental challenge suggests that even with linguistic enhancements, current architectures struggle with long-range discontinuous dependencies.

**Type Prediction Limitations.** Our binary token-level formulation predicts only span boundaries without MWE type labels, creating fundamental constraints on type-specific evaluation. While we can compute type-specific recall by checking which gold MWEs of each type were correctly identified, type-specific precision is impossible to calculate. When the model predicts a span like *systemic risk* as an MWE (false positive), we have no type prediction to determine whether the model believes this is a NOUN, VERB, or MOD/CONN expression—we only know it identified these tokens as forming some MWE. Without type-specific precision, F1 scores per type cannot be computed. This limitation means Table 2 reports only recall metrics, preventing complete type-wise performance analysis. For instance, while we know the model correctly identifies 66.9% of VERB expressions (recall), we cannot determine what percentage of its verb-like predictions are actually correct (precision) because the predictions lack type labels. This design choice prioritizes boundary detection accuracy over categorization but limits practical applications requiring typed MWE identification and restricts our ability to analyze whether false positives tend to be specific types of spurious expressions.

**Knowledge Source Dependencies.** Our linguistic features rely on spaCy's accuracy for NP chunking and dependency parsing. Errors in these upstream components propagate through our pipeline—incorrect chunk boundaries or erroneous dependency trees directly impact model performance.

**Computational Requirements.** Experiments were conducted on NVIDIA GH200 hardware, which may limit reproducibility for researchers without access to high-end GPUs. While training requires only 10-25 minutes per model, the hyperparameter search with multiple trials across

multiple oversampling ratios demands substantial compute. Our fastest configuration still requires ∼4GB GPU memory, potentially excluding older hardware.

**Language Coverage.** Evaluation is restricted to English, despite MWEs being a pervasive multilingual phenomenon with language-specific characteristics. The syntactic features we exploit (NP chunks, dependency paths) may transfer poorly to languages with different syntactic structures or morphological complexity.

## References


Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In *Handbook of Natural Language Processing*, pages 267–292. CRC Press.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Silvio Cordeiro, Carlos Ramisch, and Aline Villavicencio. 2016. LIF&UFRGS at SemEval-2016 Task 10: Rule-based MWE identification and predominant-supersense tagging. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 910–917. Association for Computational Linguistics.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *24th European Conference on Artificial Intelligence (ECAI 2020)*, pages 2006–2013. IOS Press.

Hao Fei, Yafeng Ren, and Donghong Ji. 2021. Boundary enhanced neural span classification for nested named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9016–9023.

Spence Green, Marie-Catherine de Marneffe, and Christopher D. Manning. 2013. Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1):195–227.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. NetworkX: Network analysis in Python. In *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, pages 11–15.

Haibo He and Edwardo A. Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in Python.

Yusuke Ide, Joshua Tanner, Adam Nohejl, Jacob Hoffman, Justin Vasselli, Hidetaka Kamigaito, and Taro Watanabe. 2025. CoAM: Corpus of all-type multiword expressions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27004–27021, Vienna, Austria. Association for Computational Linguistics.

Yongliang Li, Xinyu Ma, Xiaoqing Zhou, and Jizhong Qi. 2023. T2-ner: A two-stage span-based framework for unified named entity recognition with templates. *Transactions of the Association for Computational Linguistics*, 11:224–240.

Dekang Lin. 1999. Automatic identification of non-compositional phrases. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 317–324.

Alexis Nasr, Carlos Ramisch, José Deulofeu, and André Valli. 2015. Joint dependency parsing and multiword expression tokenization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1116–1126.

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, and 1 others. 2018. Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240. Association for Computational Linguistics.

Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk Abigail Walsh, Marie Candito, Ashim Gupta Mehmet Kurfalí, Verginica Barbu Mititelu, Gözde Berk, Johanna Monti, and 1 others. 2020. Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118. Association for Computational Linguistics.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, pages 82–94. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

Omid Rohanian, Shiva Taslimipoor, Samaneh Kouchaki, Le An Ha, and Ruslan Mitkov. 2019. Bridging

the gap: Attending to discontinuity in identification of multiword expressions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2692–2698, Minneapolis, Minnesota. Association for Computational Linguistics.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *International conference on intelligent text processing and computational linguistics*, pages 1–15. Springer.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain. Association for Computational Linguistics.

Nathan Schneider, Dirk Hovy, Anders Johannsen, and Marine Carpuat. 2016. SemEval-2016 Task 10: Detecting minimal semantic units and their meanings (DiMSUM). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 546–559, San Diego, California. Association for Computational Linguistics.

Nathan Schneider, Jakob Prange, Emily Danchik, Luke Gessler, Samantha Onuffer, Nora Mordowanec, Nour Kazour, and Aviram Nguyen. 2018. Comprehensive supersense disambiguation of English prepositions and possessives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1853–1865. Association for Computational Linguistics.

Raghavendra Swaminathan and Paul Cook. 2023. Token-level identification of multiword expressions using pre-trained multilingual language models. In *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, pages 1–6, Dubrovnik, Croatia. Association for Computational Linguistics.

Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. 2020. Boundary enhanced neural span classification for nested named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9016–9023.

Joshua Tanner and Jacob Hoffman. 2023. MWE as WSD: Solving multiword expression identification with word sense disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 181–193. Association for Computational Linguistics.

Kevin Walsh, Carlos Ramisch, Agata Savary, and Silvio Cordeiro. 2018. A dataset of verbal multiword expressions annotated with syntactic variants. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 101–110. Association for Computational Linguistics.

Eric Wehrli. 2000. Incremental parsing and multiword expression. In *Proceedings of the Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40.

# A Implementation and Performance Details

## A.1 Threshold Values and Hyperparameters

Table 4 presents the optimal threshold values obtained through grid search on the development set for each model configuration. The thresholds control the sensitivity of START, END, and INSIDE predictions during MWE reconstruction. Lower thresholds generally increase recall at the cost of precision.

| Model | $\tau_{start}$ | $\tau_{end}$ | $\tau_{inside}$ |
|---|---|---|---|
| BBS | – | – | – |
| BBT | 0.45 | 0.20 | 0.20 |
| BLT | 0.60 | 0.20 | 0.20 |
| DBT | 0.45 | 0.60 | 0.20 |
| DLT | 0.50 | 0.20 | 0.20 |
| DBT+l | 0.30 | 0.20 | 0.20 |
| DLT+l | 0.50 | 0.30 | 0.20 |
| DBT+o | 0.20 | 0.20 | 0.20 |
| DLT+o | 0.50 | 0.60 | 0.40 |
| DBT+a | 0.50 | 0.60 | 0.30 |
| DLT+a | 0.40 | 0.40 | 0.20 |
| DBT+lo | 0.30 | 0.50 | 0.40 |
| **DLT+lo** | **0.50** | **0.60** | **0.20** |
| DBT+la | 0.60 | 0.40 | 0.20 |
| DLT+la | 0.60 | 0.40 | 0.20 |

Table 4: Optimal threshold values for each model configuration. BBS uses span-based prediction and does not require thresholds. Best model (DLT+lo) shown in bold.

All models were trained with AdamW optimizer and linear learning rate scheduling. The best performing model (DLT+lo) used: learning rate 3e-5, batch size 16, dropout 0.3, weight decay 0.01, and early stopping with patience 3.

## A.2 Training Dynamics and Model Performance

Table 5 shows training and test performance across all configurations. The train-test gaps (17.0-30.7% F1) are expected given the limited training data (780 sentences) and the complexity of distinguishing idiomatic from compositional expressions. Notably, our best model maintains 69.8% test F1 on unseen data—the test set contains MWE patterns and combinations not present in the training data—validating our architectural choices despite the challenging dataset size.

| Model | Train F1 (%) | Test F1 (%) | Gap (Δ) |
|---|---|---|---|
| BBS | 70.7 | 53.7 | 17.0 |
| BBT | 89.6 | 58.9 | 30.7 |
| BLT | 90.1 | 59.5 | 30.6 |
| DBT | 83.2 | 63.7 | 19.5 |
| DLT | 88.9 | 62.6 | 26.3 |
| DBT+l | 92.3 | 65.9 | 26.4 |
| DLT+l | 86.1 | 59.8 | 26.3 |
| DBT+o | 89.3 | 63.4 | 25.9 |
| DLT+o | 88.7 | 62.4 | 26.3 |
| DBT+a | 86.5 | 63.0 | 23.5 |
| DLT+a | 89.6 | 64.9 | 24.7 |
| DBT+lo | 94.1 | 65.0 | 29.1 |
| **DLT+lo** | **95.0** | **69.8** | **25.2** |
| DBT+la | 93.9 | 65.1 | 28.8 |
| DLT+la | 94.6 | 68.2 | 26.4 |

Table 5: Train vs test F1 scores across all configurations. Our best model (DLT+lo) achieves 95.0% F1 on training data but maintains strong 69.8% on test data.

Figure 1 illustrates the training dynamics of our best model. The plateauing development performance after epoch 5, while training loss continues decreasing, demonstrates effective early stopping preventing overtraining.

# B Error Analysis

This appendix analyzes our best model's (DLT+lo) predictions on the test set to understand both its strong performance (69.8% F1) and areas for improvement. The analysis reveals systematic patterns in the model's decisions, demonstrating that it has learned meaningful linguistic regularities rather than arbitrary memorization.

## B.1 Overall Performance Patterns

The model generates 387 predictions for 381 gold MWEs, achieving well-balanced precision (69.3%) and recall (70.3%). Notably, 359 of 531 gold sentences (68.9%) are predicted perfectly, indicating robust pattern learning. Among the 162 imperfect sentences out of 531 gold ones, 87 contain only false negatives (missed MWEs), 51 contain only false positives (spurious detections), and 24 show both error types, typically representing partial recognition where the model correctly identifies MWE presence but misaligns boundaries.

The prediction distribution reveals interesting patterns: 329 continuous and 58 discontinuous predictions against 338 continuous and 43 discontinuous gold MWEs. While the continuous distribution aligns well, discontinuous predictions show higher variance, with 43 false positives (36.1% of all FPs) despite discontinuous MWEs comprising

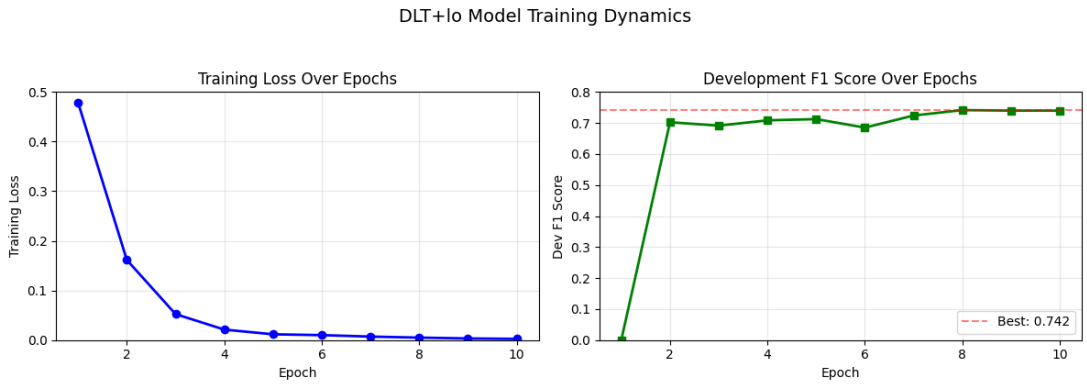


Figure 1: Learning curves for DLT+lo: training loss decreases while development F1 plateaus at epoch 8 (approximately 0.74), triggering early stopping. The initial fluctuation (epochs 2-3) reflects the model adapting to the challenging task of distinguishing genuine MWEs from compositional phrases.

only 11.3% of the gold standard.

### B.2 Continuous MWE Recognition

Continuous expressions achieve strong 74.9% recall (253/338 correctly identified). This 74.9% recall represents a significant achievement, correctly identifying three-quarters of continuous MWEs despite training on only 780 sentences. Performance varies systematically by expression type:

**High-confidence categories**: Discourse markers (*In fact*, *Of course*, *According to*) and phrasal verbs (*took over*, *brought down*, *turn out*) achieve near-perfect detection when components are adjacent. Fixed nominals like *Prime Minister* (correctly identified in all test instances) and *exchange rate* demonstrate reliable institutional term recognition.

**False positives (76 instances)**: These split into interpretable categories. Technical overdetection (23 cases) flags domain bigrams like *systemic risk* as MWEs—arguably reasonable given their specialized usage. Incomplete spans (19 cases) systematically miss required elements, such as detecting *open the way* without *for*.

**False negatives (85 instances)**: The model fails to detect MWEs that appear infrequently in training (<7 occurrences). These include: (1) metaphorical phrases like *watershed moment* and *rose-tinted glasses* that require semantic understanding beyond surface patterns, and (2) short functional MWEs like *act on* and *yet again*, where the individual words (e.g., "act", "on") appear frequently throughout the corpus in non-MWE contexts, making it difficult for the model to recognize when they specifically form a MWE.

### B.3 Discontinuous MWE Progress

Discontinuous detection demonstrates meaningful progress with 34.9% recall (15/43), doubling baseline performance and achieving 25.9% precision.

**Successful patterns**: The 15 correct predictions follow learnable structures: (1) verb-particle with single interveners like *turn [the situation] into* where dependency distance $\leq$ 3, (2) possessive frames like *on [their] own*, and (3) simple modifier insertions like *free [X] markets*.

**Error analysis**: The 43 discontinuous false positives primarily result from independent START/END predictions creating semantically invalid spans (*not...risk* spanning 8 tokens) or partial captures missing critical elements (*turning...on* without *gun*). The 28 false negatives often exceed our dependency distance threshold, like *called...back* with an 11-token gap, or involve multiple discontinuities our architecture cannot represent.

### B.4 Type-Specific Performance

| Type | Rec% | N |
|---|---|---|
| CLAUSE | 85.7 | 7 |
| MOD/CONN | 84.7 | 111 |
| VERB | 66.9 | 139 |
| NOUN | 66.9 | 121 |

Table 6: Type-specific recall performance across MWE categories for our best model (DLT+lo).

The model demonstrates strong performance across different MWE types. Formulaic expressions (MOD/CONN) achieve excellent 84.7% recall, successfully identifying phrases like *In fact* and *According to* that follow consistent syntactic

patterns. CLAUSE expressions reach 85.7% recall despite having only 7 test instances, demonstrating the model's ability to generalize from limited examples.

VERB and NOUN categories both achieve 66.9% recall, a notable accomplishment given their inherent challenges. VERB expressions require distinguishing idiomatic uses (e.g., *take off* meaning "succeed") from compositional ones (physically removing something). NOUN expressions span from technical terms to creative compounds, requiring the model to recognize both conventional phrases (*stock market*) and novel combinations.

Overall, the error analysis reveals that our model's mistakes are systematic and interpretable, often reflecting genuine linguistic ambiguities (e.g., compositional vs. idiomatic uses) rather than random failures.