Natural Language Processing and News Polysemy

Mohamed Ahmed Mansour

August 1, 2025

Text Preprocessing

- **Text Cleaning** — Text cleaning involves removing unnecessary characters and formatting issues, such as punctuation, numbers, or extra spaces. This step improves the quality of the input data for downstream tasks like tokenization or vectorization.

- **Tokenization** — Tokenization is the process of splitting text into smaller units like words or subwords. It is crucial for converting unstructured text into analyzable units. Common types of tokenization include:
  - **Sentence Tokenization**: Splits a document into individual sentences. Useful for tasks like summarization or machine translation.
  - **Word Tokenization**: Splits sentences into individual words or tokens. This is the most common form of tokenization used in NLP pipelines.
  - **Character Tokenization**: Splits text into individual characters. Used in tasks like character-level language modeling or when dealing with unknown words.

- **Stopword Removal** — Stopwords are common words like “and”, “the”, and “is” that do not contribute significant meaning. Removing them helps focus the model on more meaningful content.

- **Stemming** — Stemming is the process of reducing a word to its root or base form by removing suffixes. It uses rule-based methods without considering the context or the actual meaning of the word. For example, “playing”, “played”, and “player” may all be reduced to “play”. Stemming is faster but often less accurate than lemmatization, which uses vocabulary and morphological analysis.

- **Lemmatization** — Lemmatization reduces words to their base or dictionary form (e.g., “running” to “run”) using linguistic analysis. It is preferred over stemming for better accuracy.

- **Comparison between Stemming and Lemmatization**

Aspect	**Stemming**	**Lemmatization**
Definition	Removes suffixes to reach the root form of a word using heuristics	Reduces a word to its base or dictionary form using linguistic rules
Method	Rule-based (often using crude chopping)	Vocabulary-based + morphological analysis
Accuracy	Lower; can produce non-real words	Higher; returns actual valid words
Example	"studying", "studies" → "studi"	"studying", "studies" → "study"
Speed	Faster	Slower
Use Cases	When speed is more important than precision	When grammatical correctness and context matter

Table 1: Comparison between Stemming and Lemmatization

- **POS Tagging** — Part-of-speech tagging assigns a grammatical category to each token (e.g., noun, verb). It is used in advanced NLP tasks like parsing and named entity recognition.

Word Representation

- **Encoding**
  - – **One-Hot Encoding** — Transforms categorical variables into binary vectors. Each word is represented as a vector with one '1' at the index of that word and '0's elsewhere.
  - – **Label Encoding** — Converts categorical labels (words) into integer values. Each word gets assigned a unique numeric ID. It's useful in preparing categorical data for machine learning models.

Aspect	One-Hot Encoding	Label Encoding
Definition	Represents words as binary vectors with a single high bit	Assigns each word a unique integer value
Output Format	Binary matrix	Integer vector
Interpretability	Easy to interpret	Less intuitive for categorical comparisons
Sparsity	High (mostly zeros)	Low
Use Cases	Suitable for non-ordinal categories	Suitable for tree-based models

Table 2: Comparison between One-Hot and Label Encoding

- **Bag of Words (BoW)** — Represents text as a vector of word occurrence counts, disregarding grammar and word order but keeping multiplicity.
- **TF-IDF (Term Frequency-Inverse Document Frequency)** — TF-IDF adjusts the word counts in BoW based on how frequently words appear across documents. It reduces the importance of common words.
- **N-Grams** — N-grams are contiguous sequences of n words (e.g., unigrams = 1, bigrams = 2). They capture local word context in the text.
- **Occurrence Matrix** — A matrix showing how many times each word appears in each document. Each row is a document; each column is a vocabulary term.
- **Co-occurrence Matrix** — Measures how often words appear together in the same context (e.g., same document). Useful for understanding semantic relationships between words.

Embedding

Embedding Techniques

- **Word Embedding:** A method to represent words in dense vector space, capturing semantic meaning and relationships.
- **Word2Vec:** Predictive embedding model by Google using Skip-Gram or CBOW architectures to learn word associations from large corpora.
- **GloVe:** (Global Vectors for Word Representation) by Stanford. It combines global matrix factorization and local context windowing to generate word vectors.
- **FastText:** Developed by Facebook, it improves on Word2Vec by considering subword information, allowing better handling of rare and out-of-vocabulary words.
- **ELMo:** (Embeddings from Language Models) Provides context-aware word representations using deep bidirectional LSTMs trained on a language modeling task.

- **BERT:** (Bidirectional Encoder Representations from Transformers) A transformer-based model that provides contextualized embeddings using masked language modeling.

Comparison: Word2Vec vs GloVe

Feature	Word2Vec	GloVe
Training Method	Predictive (learns by predicting context words)	Count-based (factorizes word co-occurrence matrix)
Context Window	Local	Global
Computational Efficiency	Faster on large corpora	Requires more preprocessing (matrix factorization)
Accuracy	High	Comparable or slightly better for semantic tasks

1 Dual-Branch Fake News Detection Framework

1.1 System Overview

The goal of this framework is to classify whether a news article is real or fake using two distinct branches:

- **Text Branch (BERT):** Understands the semantics and context of the article.
- **Knowledge Branch (TransE):** Matches extracted factual information against a structured knowledge graph.

Main Components:

- **Triplet Extraction:** Uses REBEL to generate (Head, Relation, Tail) triplets from text.
- **Fuzzy Matching:** Aligns extracted triplets to entries in the knowledge graph using Levenshtein distance.
- **Dual Branches:** BERT encodes text semantics; TransE scores triplet logic.
- **Interaction Module:** Fuses scores from both branches to produce the final prediction.

1.2 Triplet Extraction with REBEL

REBEL (Relation Extraction By End-to-end Language generation) based on BART is used to extract structured triplets from text.

Example: *"Einstein invented the internet"* $\rightarrow$ (Einstein, invented, internet)

1.3 Fuzzy Matching with Knowledge Graph

To verify factual consistency, triplets are matched to an external knowledge graph (e.g., CSKG) using Levenshtein distance to compute similarity between triplet elements and knowledge base entries.

1.4 Text Branch: BERT-Based Semantic Encoder

This branch tokenizes and encodes the article using BERT to produce contextual embeddings representing the document's overall meaning.

Output: Semantic vector V_D.

1.5 Knowledge Branch: TransE Triplet Embedding

- **Triplet Aggregation Module:** Each triplet is embedded using TransE, and the resulting embeddings are aggregated with MLP-Mixer to form a document-level knowledge vector Y_D.
- **Triplet Scoring Module:** Each triplet is scored using the formula:

$$score(h, r, t) = \|h + r - t\| \Rightarrow p_{hrt} \text{ is the final rationality score}$$

1.6 Interaction Module and Final Prediction

The prediction is made by fusing the outputs of both branches:

$$p_{predict} = g \cdot p_{text} + (1 - g) \cdot p_{hrt}$$

where g is a learnable weight controlling the balance between semantic and factual inputs.

Classification Rule: If $p_{predict} >$ threshold, the article is classified as Fake News.

1.7 Loss Function

- Binary Cross-Entropy Loss is used for classification.
- L2 Regularization is applied to prevent overfitting.

1.8 Comparison Table: Text Branch vs. Knowledge Branch

Aspect	**Text Branch (BERT)**	**Knowledge Branch (TransE)**
Purpose	Context understanding	Factual verification
Input	Full text article	Extracted triplets (h, r, t)
Method	Transformer encoding	Triplet embedding and scoring
Strength	Captures semantics	Real-world logic and knowledge
Output	Semantic vector (V_D)	Rationality score (p_{hrt})

Table 3: Comparison between Text and Knowledge Branches

1.9 Fake News Detection with Semantic Triplets

This dual-branch framework integrates:

- **Contextual Branch:** Based on BERT.
- **Knowledge Branch:** Based on TransE and aligned triplets from CSKG.

1.10 Datasets Used

- **FakeNewsNet:** Built from PolitiFact and GossipCop. Includes news content and social context (tweets, user profiles).
- **ReNews:** Constructed by the authors. Provides labeled articles with REBEL-extracted triplets matched to CSKG.

1.11 Implementation Tools and Settings

- **Libraries:** Hugging Face Transformers, PyTorch, OpenKE.
- **Configuration:** AdamW optimizer, 10 epochs, batch size 32, Binary Cross Entropy loss with L2 regularization.
- **Hardware:** NVIDIA V100 GPU.

1.12 Evaluation Metrics

- Accuracy, Precision, Recall, F1-score, AUC-ROC.

1.13 Model Comparison Results

Model	Accuracy	F1-Score
FakeBERT	91.38%	91.32%
NewsGraph	92.61%	92.30%
FND-SCTI (Proposed)	95.83%	95.47%

Table 4: Performance on ReNews Dataset

Model	Accuracy	F1-Score
FakeBERT	87.75%	87.34%
NewsGraph	88.94%	88.23%
FND-SCTI (Proposed)	90.55%	90.41%

Table 5: Performance on FakeNewsNet Dataset

1.14 Baseline Models Used

Text-Based Baselines: BiGRU, TextCNN, FakeBERT.

Social Context Models: SAFE, GDU.

Knowledge Graph-Based: NewsGraph.

Proposed Method: FND-SCTI (BERT + TransE).

1.15 Ablation Study

To evaluate the contribution of each module:

- **Without Triplet Module:** Removes REBEL + CSKG.

- **Without External Knowledge:** No CSKG alignment.
- **Text-Only:** Only BERT is used.

Result: Removing semantic triplet knowledge significantly harms performance.

1.16 Conclusion and Future Work

- Introduced a dual-branch architecture combining linguistic and factual reasoning.
- Demonstrated superior performance on ReNews and FakeNewsNet.
- Future directions: handle multi-modal data (e.g., images/videos), enable real-time fake news detection pipelines.

2 Polysemy in the News

2.1 Introduction to Polysemy in the News

- News stories can have multiple meanings depending on how they are written and who reads them.
- The article focuses on how different types of news reporting create agreement or disagreement among readers.
- It uses the example of Dan Price, a CEO who raised his employees' minimum salary, to explore how the same story can be interpreted differently.

2.2 Corporate Social Responsibility (CSR) and Media

- CSR is when companies do good things for society beyond making money (e.g., fair wages, environmental care).
- The Dan Price case is a CSR example: he cut his own salary to raise employees' wages.
- The media responded in different ways — some praised him, others doubted his motives.

2.3 Dan Price: The Ethical Capitalist?

- Dan Price became famous for his salary decision in 2015.
- Motivated by fairness and research showing that happiness plateaus after a certain income.
- Seen by some as a hero and others as unrealistic or performative.

2.4 Methodology

- The author collected 9 news articles and 6000 Facebook comments.
- Surveys were conducted with 1000+ participants via Amazon Mechanical Turk.

- Articles were categorized into:
  - Convergent: Where most people agreed (low polysemy).
  - Divergent: Where people disagreed (high polysemy).

2.5 Comparing Convergent and Divergent Articles

- **Convergent Articles**
  - Present Dan as a hero.
  - Use emotional, personal storytelling.
  - Use flattering images and focus on success.
  - End on positive notes.
- **Divergent Articles**
  - Ask critical questions and present multiple sides.
  - Include expert opinions and statistics.
  - Present neutral or skeptical tone.
  - Use less emotional language and fewer flattering images.

Aspect	**Convergent Articles**	**Divergent Articles**
Character portrayal	Heroic, emotional	Neutral or questioning
Conflict framing	Mythic, moral battle	Data-driven debate
Ending style	Positive closure	Open-ended or uncertain
Tone	Emotional, affirming	Analytical, cautious

Table 6: Comparison of Convergent vs Divergent Articles

2.6 Mechanisms of Polysemy in Journalism

- **Character Construction** — Convergent texts create heroes; divergent texts present complex, less defined characters.
- **Conflict Framing** — Convergent texts present moral clarity; divergent texts encourage debate.
- **Conclusion Style** — Convergent stories offer satisfying endings; divergent stories remain open-ended.

2.7 Cultural and Stylistic Influence on Interpretation

- People from different cultures may interpret the same story differently.
- Styles like podcasts or cliffhangers increase polysemy because they encourage personal interpretation.

- Topics like wealth inequality, fairness, and justice inherently provoke multiple viewpoints.

2.8 Conclusion and Implications

- News is more than just facts — style and structure affect meaning.
- Polysemy is a natural result of diverse interpretations and storytelling styles.
- Journalists have the power to influence reader perception by how they frame stories.