Semantic Analysis and Role Labeling in News Articles

DINESH RAM DANARAJ
NISCHITH BAIRANNANAVARA OMPRAKASH
VISHAL PRAKASH
MANIKUMAR HONNENAHALLI LAKSHMINARAYANA SWAMY
NANDAN DAVE

Abstract

Our research focuses on advancing natural language processing by combining Semantic Role Labelling (SRL) with cutting-edge NLP techniques. By analyzing news articles, we aim to understand the semantics of text better. We improved the training dataset by refining the Knowledge Extraction from Language Model (KELM) dataset and the Bloomberg Quint news dataset using spaCy. This refinement enhanced the precision of our dataset for fine-tuning the BERT model. Our paper presents a detailed approach that utilizes BERT for semantic role labeling. This involves predicting actors and actions within entire news articles, which significantly enhances contextual understanding. Additionally, we introduce an Impact Score method that integrates AFINN ratings, textblob, vader, and transformer pipelines to quantify the relevance of actor-action pairs. Our results showcase the effectiveness of our refined annotations in creating more coherent and enriched narratives. This is crucial for achieving accurate and efficient semantic role labeling, ultimately leading to a better understanding of text semantics.

1 Introduction

Semantic Role Labeling (SRL) plays a crucial role in the advancement of natural language processing (NLP) by facilitating a deeper semantic understanding of text through the identification of predicate-argument structures. This research paper explores the application of SRL integrated with advanced NLP methodologies to enhance the extraction and categorization of actor-action pairs from news articles. Our approach simulates a more sophisticated, human-like interpretation of textual data, which is essential for improving information retrieval, content summarization, and automated insights generation. Employing the KELM corpus and the Bloomberg Quint news dataset, we refine initial data sets using spaCy's capabilities in Named Entity Recognition, Dependency Parsing, and Neural Coreference Resolution to prepare inputs for a BERT model adapted specifically for semantic parsing tasks. The expected outcome is a significant improvement in model performance, particularly in its ability to discern and understand complex narrative structures. By pushing the boundaries of current NLP applications, this study aims to contribute to the development of more autonomous systems capable of managing and analyzing large volumes of textual information with minimal human oversight. The broader implications of this work include enhanced accuracy in news summarization and more effective monitoring and reporting systems, which are increasingly crucial in the digital information age.

2 Related Work

Semantic analysis and Role Labelling is an important and extensively studied field in the field of natural language processing. It is used for better understanding of textual data which allows for text summarization, machine translation, sentiment analysis etc. This section focuses on contributions in the field of Semantic Role Labelling.

2.1 Semantic Role Labelling for News Tweets

(Liu et al., 2010) applied semantic role labelling to news tweets. News tweets are characterized by their brevity, informal language, syntactic irregularities and frequent use of colloquial expression. The authors have not only taken a big step forward in applying semantic role labelling to informal texts, but also in enhancing information extraction and retrieval capabilities in real time media context.

2.2 Advancements in Semantic Role labelling

Aside from the context of news tweets, semantic role labelling has seen a lot of development. This is shown by the work of (Gildea and Jurafsky, 2002) who used FrameNet and PropBank for automatic annotation of semantic roles. These also provides structured, verb-cantered annotations that helps in understanding predicate argument structures.

The advancement of deep learning also brought significant advancements in semantic role labelling. (He et al., 2017) created a deep bidirectional LSTM model which achieved state of the art results. The model used sentence level representations without extensive feature engineering to achieve this. It enabled context embedding that captured nuanced meanings.

(Devlin et al., 2018) introduces Bidirectional Encoder Representations from Transformers (BERT). The model is pre-trained using deep bidirectional representations by conditioning on both left and right textual contexts across all layers. This allowed it to capture a more nuanced understanding of language syntax and semantics. This allows it to be used for wider range of tasks.

(Arora et al., 2020) analysed conditions under which deep contextual embedding with BERT outperforms simpler baselines such as GloVe and random word embedding. They found that deep contextual embedding work better on tasks involving complex linguistic structures, ambiguous terminology, novel word usage while simpler pre-trained models are move useful in in analysing large, industry scale datasets.

(Tian et al., 2022a) has proposed map memories to improve semantic role labelling by encoding different types of auto-generated syntactic knowledge like POS tagging, syntactic constituencies and word dependencies.

3 Methods

3.1 Data Collection

In the data collection phase of our research, we concentrated on refining the initial dataset to precisely extract actor-action pairs, which are essential for deep semantic analysis. Our starting point was the KELM corpus, which, although rich in information, was encumbered with superfluous details. For example, the dataset included entries like "triples": [["Heyward", "won", "election"]] associated with sentences such as "Heyward won in the runoff election and became the 88th governor of South Carolina." Such verbosity was not necessary for our analysis needs and thus required streamlining.

To enhance the usability of this data, we employed a Convolutional Neural Network (CNN) model from spaCy, a state-of-the-art tool renowned for its advanced natural language processing (NLP) capabilities. We utilized spaCy's Named Entity Recognition (NER), Dependency Parsing, and Neural Coreference Resolution functionalities to parse complex sentences effectively. This allowed us to distill the essential information from the verbose initial data.

For instance, the application of our spaCy model transformed the detailed initial entry into a more streamlined actor-action format: "Heyward": ["win", "become"]. This revised format succinctly captured the key semantic actions, indicating that "Heyward" not only "won" the election but also "became" the 88th governor. This process of refinement was crucial in removing irrelevant details and restructuring the data into a format that was both concise and highly relevant to our analysis objectives.

This meticulous extraction of actor-action pairs resulted in a well-defined dataset that was optimal for conducting semantic role labeling. The clean and precise dataset significantly enhanced the accuracy of our semantic analysis, ensuring that our study was built on a robust foundation of clearly defined semantic roles. This streamlined dataset not only facilitated a more efficient analytical process but also improved the overall quality and interpretability of our research findings.

3.2 Data Annotations

Prior to initiating the annotation process for BERT pre-training, we engage in a series of preliminary steps designed to refine the text data. These steps—Name Concatenation, Dependency Parsing, and Coreference Resolution—are critical in transforming unstructured text into a format that can be effectively used for annotating and training machine learning models, such as BERT.

3.2.1 Name Concatenation

To streamline the dataset for semantic analysis, we utilised a name concatenation method to handle multi-token proper nouns and named entities effectively. This process involved merging contiguous tokens that constitute a single proper noun or named entity, thereby reducing data sparsity and improving model performance. For example, in the dataset, entities like "New York" or "Prime Minister" are often split into multiple tokens. Our concatenation script identifies these multi-token entities using B/I/O tagging and merges them into

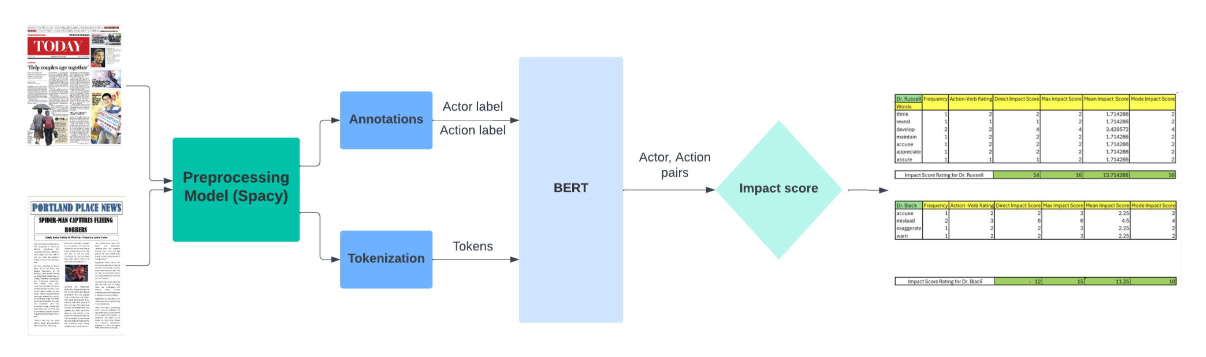

Figure 1: End-to-end Flow Diagram

single tokens, "NewYork" and "PrimeMinister", ensuring that they are treated as single units in subsequent processing stages.

3.2.2 Dependency Parsing

SpaCy's dependency parsing is utilized to identify the subjects and objects associated with each verb within sentences (Tian et al., 2022b). This method allowed us to establish connections between verbs and all pertinent actors. By concentrating on verb-centered tuples and implementing filters based on parts of speech and dependency tags (such as nominal subject and direct object), we were able to effectively capture the fundamental interactions within sentences that are pivotal for comprehending the narrative context.

For example, when applying this approach to the sentence "Emma and Sophia visited Paris and London, while James and Ethan explored New York and Tokyo," we would obtain the following output:

“visited” :[[“Emma”, “Sophia”], [“Paris”, “London”]] (verb : “visited”, subject: [“Emma”, “Sophia”] and object: [“Paris”, “London”])

“explored” : [[“James”, “Ethan”], [“NewYork”, “London”]] (verb : “explored”, subject: [“James”, “Ethan”] and object: [“NewYork”, “Tokyo”])

3.2.3 Coreference Resolution

Coreference resolution aims to resolve pronouns to respective entity within the text. Utilising the neural correlation capabilities of SpaCy, we resolve references such as "he", "they" to their respective entities, like "John" or "NewYork". This resolution is crucial for maintaining narrative consistency across sentences and is particularly valuable in mapping the actual actor to the verb instead of just mapping the pronouns. For example, in the text "At the zoo, John, Mary, and Sarah observed the animals in their habitats. While they were watching the monkeys, John pointed out that he had seen one of them at the zoo before.", "their" would resolve to "animals", "they" would resolve to "John", "Mary", and "Sarah" and "he" would resolve to "John". This resolution not only enhances the clarity of the narrative but also supports more accurate semantic role assignments by maintaining referential integrity.

To train BERT for the nuanced task of semantic role labelling, the input data must meticulously delineate the intricate interplay between actors and their corresponding actions. This is achieved through a structured annotation process that builds upon a foundational, contextually-enriched framework, which is essential for the fine-tuning of the model. Consider the sentence: “SteveJobs founded the company Apple” In preparation for BERT training, this sentence is dissected into distinct tokens: [’SteveJobs’, ’founded’, ’the’, ’company’, ’Apple’]. Each token is then assigned an actor vector and an action vector each of which is of dimension 1X20, assuming there are no mo more then 20 actors per sentence.
Actor Vector Assignment: The token ’SteveJobs’ is identified as the main actor, marked by an actor vector [1, 0, 0, ..., 0], signifying its primacy. Similarly, ’Apple’ is recognized as another key entity within the narrative, denoted by the actor vector [0, 1, 0, ..., 0].
Action Vector Assignment: The verb token ’founded’ establishes the pivotal action connecting ’SteveJobs’ and ’Apple’. This relationship is encoded in an action vector [1, 1, 0, ..., 0], linking the action directly to both entities. Other tokens

in the sentence that do not convey action possess a default vector [0, 0, 0, ..., 0], reflecting their non-action roles within the semantic structure.
For sentence “SteveJobs founded the company Apple”, the detailed annotation string that will be fed to BERT:
actors vector = [[1, 0, . . . , 0, 0], [0, 0, . . . , 0, 0], [0, 0, . . . , 0, 0], [0, 0, . . . , 0, 0], [0, 1, . . . , 0, 0]]
actions vector = [[0, 0, . . . , 0, 0], [1, 1, . . . , 0, 0], [0, 0, . . . , 0, 0], [0, 0, . . . , 0, 0], [0, 0, . . . , 0, 0]]

The binary vectors facilitate BERT’s comprehension of the semantic roles, enabling it to predict with precision the actor-action dynamics present in novel sentences. This process ensures that BERT is not merely capturing the syntactic representation of text but is also adept at interpreting and reproducing the complex semantic patterns that it learns during training.

3.3 Model

3.3.1 BERT

To leverage the bidirectional capability of the BERT model (Devlin et al., 2018), pre-processed and annotated sentences are used to fine-tune the BERT model to perform multi-label token classification, identifying and classifying actors and their actions within sentences. The system assigns each token to any of 20 unique labels, divided evenly between 10 actor categories (actor1, actor2, ... actor10) and 10 labels corresponding to the actions in which actors are involved (actor1-action, actor2-action, ... actor10-action). The bidirectional capabilities of BERT enable this model to effectively interpret complex textual contexts and accurately identify both actors and their corresponding actions. (Shi and Lin, 2019)

A dataset consisting of 1500 pre-processed and annotated sentences is used to fine-tune the pretrained BERT model (roberta-base) for this multi-label classification task, optimizing the training with a batch size of 4 across 10 epochs with BCEWithLogits as the loss function to efficiently enhance performance. By applying a threshold of 0.35 for actor labels and 0.25 for action label to the sigmoid-activated outputs, we binarize the predictions to accurately identify relevant tokens for each label. This approach ensures that the model achieves high granularity in token-level predictions, making it ideal for detailed semantic analyses in automated text processing environments.

Given the input sentence "John killed Jack using a knife." the model decomposed the sentence to tokens ["John", "Killed", "Jack", "using", "a", "Knife", "."] and output the below actor and action labels for each token.

"John": Actor-Labels=[1, 0, 0, 0, . . . , 0], Action-Labels= [0, 0, 0, 0, . . . , 0] (John is actor1)

"Killed": Actor-Labels=[0, 0, 0, 0, . . . , 0], Action-Labels=[1, 1, 0, 0, . . . , 0] (Actor1(John) and Actor2 is involved in action word "Killed")

"Jack": Actor-Labels=[0, 1, 0, 0, 0, . . . , 0], Action-Labels=[0, 0, 0, 0, . . . , 0] (Jack is Actor2)

"using": Actor-Labels=[0, 0, 0, 0, 0, . . . , 0], Verb-Labels=[1, 1, 0, 0, . . . , 0] (Actor1 and Actor2 are involved in action word "using")

"a": Actor-Labels=[0, 0, 0, 0, . . . , 0], Action-Labels=[0, 0, 0, 0, . . . , 0]

"knife": Actor-Labels=[0, 0, 0, 0, . . . , 0], Action-Labels=[0, 0, 0, 0, . . . , 0]

".": Actor-Labels=[0, 0, 0, 0, . . . , 0], Action-Labels=[0, 0, 0, 0, . . . , 0]

3.4 Impact Score Calculation

The Impact Score is a new approach for extracting significant actors and actions from newspaper content. This section explores how integrating the frequency of occurrences(actors) with corresponding action ratings, based on average of AFINN data for action ratings, Vader Sentiment, Sentiment transformer pipeline and textblob sentiment polarity ratings, can accurately determine the relevant score.

3.4.1 Textblob polarity rating

One of the features of TextBlob is its ability to compute the polarity of a text, which is part of its sentiment analysis capabilities. Polarity is a float within the range [-1.0, 1.0] where -1.0 signifies negative sentiment and 1.0 signifies positive sentiment. It is calculated as part of TextBlob’s sentiment analysis method, which uses pattern analysis to assess the sentiment of a text. This polarity score represents the overall emotion expressed in the text.

3.4.2 Vader Sentiment Score

VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments. VADER sentiment analysis returns three components: the positive, negative, and neutral scores of the text, along with a composite score called the "compound" score, which is a normalized, weighted composite score. This score is a sum of all the lexicon ratings which have been normalized between [-1, 1], where -1 is most extreme negative and 1 is most extreme positive.

3.4.3 AFINN Action rating Score

The AFINN lexicon is a list of words rated for valence with an integer between minus five (negative) and plus five (positive). The ratings are assigned manually by Finn Årup Nielsen and are based on his sentiment about the word. This simple and straightforward approach provides a quick and easy way to carry out sentiment analysis

3.4.4 Transformer pipeline Sentiment score

Using advanced models like the Hugging Face transformers for sentiment analysis, particularly for contextually evaluating action verbs, is a sophisticated approach that leverages deep learning techniques to understand the nuances of language more effectively than simpler models.

Based on the above three different verb sentiment's score, the action rating score is assigned to each action. As the ratings range from different values, to calculate the action rating the above scores are scaled to +/-5 and then average score of these sentiment scores are taken as the action rating score for a verb.

To find the best possible combination of impact score calculation, a experiment is conducted with human evaluation of a newspaper article. Where the readers had to map the best actor with their most appropriate action from the given passage.

For the calculation four different approaches are taken to find lay the base of the calculations, i.e, direct impact score, mean impact score, mode impact score, maximum impact score.
The calculations for these 4 are taken as :
direct impact score = Action-rating of the verb * frequency
maximum impact score = max(action-rating of the verbs) * frequency
mode impact score = mode(action-rating of the verb) * frequency
mean impact score = mean(action-rating for the verb) * frequency

3.5 Real-world Application of Impact Score Calculation

In demonstrating the practical application of our proposed Impact Score (IS), we analyzed a *news article* detailing a burglary at the Los Angeles residence of actor Keanu Reeves. This incident, as reported, involved multiple intruders wearing ski masks, engaging in various actions, notably trespassing and theft.

The various impact score calculations for this article are shown in tables 1, 2, 3 and 4.

Table 1: Impact Scores (IS) for Actor 1: Keanu Reeves

Action	Frequency	Action Rating
burglarized	2	1.9973613768815994
wearing	1	1.9527792632579803
smashed	1	1.999656304717064
broke	1	2.2713828045129776
prompted	1	1.9919049441814423
respond	1	1.9972735345363617
resulting	1	1.990556612610817
reviewing	1	1.994041547179222
secured	1	2.502102979302406
use	1	1.9811984598636627
wandered	1	1.9826207011938095

Table 2: Total Impact Scores (IS) for Actor 2: Bryan Dixon

Action	Frequency	Action Rating
investigate	1	1.9813557416200638
peering	1	1.9903303384780884
smoking	1	1.9979659169912338
walking	1	1.9970093369483948
returned	1	1.9867942929267883
leaving	1	1.9990061670541763
trespassing	1	1.9889571219682693
entered	1	1.9981668591499329
fell	1	1.999566227197647
thrown	1	1.9965478479862213

The Impact Score calculations, illustrated in the appended tables, revealed nuanced insights into the actors involved and their actions' significance. Based on the frequency of occurrence and action-verb ratings derived from the AFINN sentiment lexicon, VADER, TextBlob and Sentiment Transformer pipeline, several impact score types were

Table 3: Impact Scores (IS) for Actor 3: Intruders

Action	Frequency	Action Rating
prompted	1	1.9919049441814423
respond	1	1.9972735345363617
resulting	1	1.990556612610817
wandered	1	1.9826207011938095
discovered	1	1.9979893118143082
cleaning	1	1.986114501953125
According	1	1.9939832836389542
walked	1	1.992508888244629

Table 4: Total Impact Scores of Actors

Actor	Direct IS	Max IS	Mean IS	Mode IS
Keanu Reeves	34.6582399	12.50210297	12.05485332	13.99472275
Bryan Dixon	24.9356998	6.99956622	6.99356998	6.99956622
Intruders	18.9329517	4.99798931	4.99161897	4.99798931

calculated: Direct, Max, Mean, and Mode Impact Scores.

From our analysis, we discerned that the primary actors, designated as 'Keanu Reeves', possessed the highest cumulative Direct Impact Score, indicating their predominant role in the narrative. This was substantiated by the fact that each type of impact score associated with the burglars exceeded those of other actors in the article. Additionally, examining the actions of the burglars, we identified 'Burglarised' as the action with the highest individual impact scores across all types, illustrating its centrality in the incident's description.

By leveraging the comprehensive data captured in the article, our Impact Score calculations corroborated the assertion that Keanu Reeves, with his actions, predominantly shaped the event's portrayal. Our methodology thus validated the effectiveness of Impact Scores in extracting and quantifying actor-action dynamics from textual content.

4 Result

4.1 Evaluation

4.1.1 F1 Score Evaluation for Actor and Action Label Classification

The F1 score is used as a metric to evaluate the performance of actor and action identification, a process that combines preprocessing and fine-tuned model output. Five news articles are manually annotated for actors and actions. The same news articles are preprocessed and passed through the fine-tuned model, and the actors and actions identified from the combined output are compared with the manual annotations to identify correctly identified actors and actions (True Positives), actors and actions not identified (False Negatives), and words wrongly classified as actors or actions (False Positives).

Based on these outcomes, precision and recall are calculated to evaluate the performance of the model. The F1 score for the model on the validation dataset, consisting of five articles selected from *Bloomberg Quint News Article*, yielded an average of 0.7839.

To analyze the detailed calculation of the F1 score for an article, the *Keanu Reeves article* is taken as an example, and the human-evaluated annotation and model-generated output are compared. The results for the F1 evaluation are presented in Table 5. Table 6 displays the human-evaluated annotation, and Table 7 shows the model output.

Table 5: F1 score for *Keanu Reeves Article*

Precision	Recall	F1 Score
0.78	0.835	0.8066

Table 6: Human-Evaluated Annotations

Actor	Frequency	Action
KeanuReeves	10	'burglarized', 'wearing', 'smashed', 'burglarized', 'broke', 'resulting', 'reviewing', 'secured', 'wandered'
LosAngeles	2	'burglarized', 'burglarized'
house	4	'burglarized', 'entering', 'scoping', 'showed'
JohnWick	1	
police	4	'arrived', 'received', 'discovered', 'activated', 'looking', 'investigated'
TMZ	3	'according', 'said', 'According'
intruder	3	'prompted', 'respond', 'resulting', 'wandered', 'discovered', 'walked'
LAPD	1	'prompted', 'investigation'
officers	1	'search'
trespassers	1	'discovered'
security	4	'captured', 'hired', 'showing', 'shows'
individuals	1	'wearing', 'breaking', 'entering'
robbers	1	'stole', 'making'
Investigators	1	'reviewing', 'surrounding'
cops	1	'made', 'scoping'
stalker	1	'secured', 'showed'
actor	3	'hired', 'investigate', 'investigated'
theHollywoodHills	1	'owns', 'hired', 'investigate'
BryanDixon	5	'investigate', 'peering', 'smoking', 'walking', 'returned', 'leaving', 'trespassing', 'entered', 'fell', 'thrown'
LA	1	
intruders	1	
team	1	'discovered'
Authorities	1	'arrived', 'escorted'
company	1	'cleaning', 'left'
FedEx	1	'placed'
masterbuilder	1	

Table 7: Annotations Generated by the Model

Actor	Frequency	Action
KeanuReeves	10	’burglarized’, ’wearing’, ’smashed’, ’burglarized’, ’broke’, ’prompted’, ’respond’, ’resulting’, ’reviewing’, ’secured’, ’use’, ’wandered’
LosAngeles	2	’burglarized’, ’wearing’, ’burglarized’
house	4	’burglarized’, ’entering’, ’scoping’, ’showed’
JohnWick	1	
police	4	’arrived’, ’received’, ’discovered’, ’activated’, ’looking’, ’investigated’
TMZ	3	’according’, ’said’, ’According’
intruder	3	’prompted’, ’respond’, ’resulting’, ’wandered’, ’discovered’, ’cleaning’, ’According’, ’walked’
LAPD	1	’prompted’
officers	1	
trespassers	1	’discovered’
security	4	’captured’, ’hired’, ’showing’, ’shows’
individuals	1	’wearing’, ’breaking’, ’entering’
robbers	1	’stole’, ’making’, ’according’
Investigators	1	’reviewing’, ’surrounding’
cops	1	’made’, ’scoping’
stalker	1	’secured’
actor	3	’hired’, ’investigate’, ’investigated’, ’showed’
theHollywoodHills	1	’owns’, ’hired’, ’investigate’
BryanDixon	5	’investigate’, ’peering’, ’smoking’, ’walking’, ’returned’, ’leaving’, ’trespassing’, ’entered’, ’fell’, ’thrown’
LA	1	
star	1	
intruders	1	
team	1	’discovered’
Authorities	1	’arrived’, ’escorted’
company	1	’cleaning’, ’left’
FedEx	1	’investigated’, ’placed’
world	1	’building’
Speed	1	
masterbuilder	1	

4.1.2 Human evaluation for Impact score

Correlating the impact score outcomes with evaluations made by humans on articles. An effective approach to assess our calculations is by matching our results with the real-world analysis conducted by readers of the articles. We asked 5 readers to study the Keanu Reeves article and identify the main actors and their action. All of the 5 readers identified the ***Keanu Reeves*** as the main actors and ***Buglarized*** as their main action. According to them, the top 5 actors were : Keanu Reeves, Bryan Dixon, Intruder, Police, Keanu’s house.

4.2 Comparison of Impact score with frequency and action rating for predicting major actors

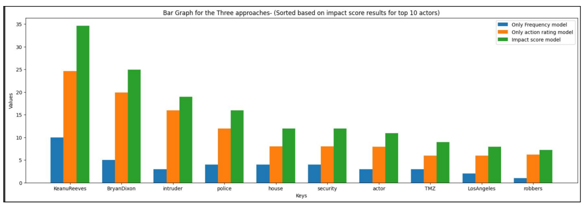

Figure 2: Comparison of Impact score with frequency and action rating

In this section, we compare the results of Keanu Reeves article with impact score model, only action rating model and only frequency model. This is shown in *Figure 2*. The Impact Score model tends to yield higher values for most actors, indicating a potentially more significant assessment for the article.

Based on the graph it is clear that the ranking for top 10 actors are different from each approach and the closest ranking to the ones given by human evaluations from section 4.1.2. is that of ranking generated from impact score approach. These ranking from impact score can be referred from Table-8.

Table 8: Impact Score and ranking of actors from the article

Actor	Ranking	Impact Score
1	KeanuReeves	34.65823990511895
2	BryanDixon	24.935699850320816
3	intruder	18.932951778173447
4	police	15.949296414852142
5	house	11.979784712195396
6	security	11.979272857308388
7	actor	10.9290781468153
8	TMZ	8.978271201252937
9	LosAngeles	7.947502017021179
10	robbers	7.239514574408531
11	theHollywoodHills	6.972695857286453
12	individuals	6.327054172754288
13	Authorities	4.996149614453316
14	cops	4.987707212567329
15	Investigators	4.986014887690544
16	company	4.985591992735863
17	FedEx	4.955524206161499
18	stalker	3.502102979302406
19	trespassers	2.997989311814308
20	team	2.997989311814308
21	world	2.9971984326839447
22	LAPD	2.9919049441814423
23	JohnWick	1
24	officers	1
25	LA	1
26	intruders	1
27	Speed	1
28	masterbuilder	1

5 Conclusion and limitation

Based on the resultant calculation, impact score can be used as a additional evaluation metric for figuring out popular searches, apart from frequency measurement or strength or rating measurement. Thus, ranking the actors with their corresponding actions based on impact score gives better results and closer to human estimation compared to the other two approaches(frequency and action score rating).

One limitation of this model is its occasional inability to recognize the author's method of referencing an actor alongside other common nouns. For example, if Keanu Reeves is denoted as "Actor" within a sentence, the model may interpret "actor" as denoting a distinct actor rather than understanding its contextual reference to Keanu Reeves.

References

Simran Arora, Avner May, Jian Zhang, and Christopher Ré. 2020. Contextual embeddings: When are they worth it? *arXiv preprint arXiv:2005.09117*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

Xiaohua Liu, Kuan Li, Bo Han, Ming Zhou, Long Jiang, Zhongyang Xiong, and Changning Huang. 2010. Semantic role labeling for news tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 698–706.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Yuanhe Tian, Han Qin, Fei Xia, and Yan Song. 2022a. Syntax-driven approach for semantic role labeling. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7129–7139.

Yuanhe Tian, Han Qin, Fei Xia, and Yan Song. 2022b. Syntax-driven approach for semantic role labeling. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7129–7139, Marseille, France. European Language Resources Association.