Information Processing and Management 59 (2022) 102957

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

A systemic functional linguistics approach to implicit entity recognition in tweets

Hawre Hosseini [a], Mehran Mansouri [b], Ebrahim Bagheri [a,*]

[a] Laboratory for Systems, Software and Semantics (LS^3) Ryerson University, Toronto, Canada
[b] Allameh Tabataba'i University, Tehran, Iran

ARTICLE INFO

Keywords:
Named entity recognition
Tweet analytics
Knowledge graphs
Computational linguistics

ABSTRACT

The identification of knowledge graph entity mentions in textual content has already attracted much attention. The major assumption of existing work is that entities are explicitly mentioned in text and would only need to be disambiguated and linked. However, this assumption does not necessarily hold for social content where a significant portion of information is implied. The focus of our work in this paper is to identify whether textual social content include implicit mentions of knowledge graph entities or not, hence forming a two-class classification problem. To this end, we adopt the systemic functional linguistic framework that allows for capturing meaning expressed through language. Based on this theoretical framework we systematically introduce two classes of features, namely syntagmatic and paradigmatic features, for implicit entity recognition. In our experiments, we show the utility of these features for the task, report on ablation studies, measure the impact of each feature subset on each other and also provide a detailed error analysis of our technique.

1. Introduction

In the context of textual social content such as tweets, the task of explicit named entity recognition and linking has already been addressed with reasonable performance (Ferragina & Scaiella, 2010; Ibrahim, Amir Yosef, & Weikum, 2014; Meij, Weerkamp, & De Rijke, 2012; Nozza, Manchanda, Fersini, Palmonari, & Messina, 2021) and applied in the context of other tasks (Das & Paik, 2021; Mendoza, Parra, & Soto, 2020; Zhao et al., 2021) despite the challenges that short social content pose due to their anti-grammatical and noisy nature (Botzer, Ding, & Weninger, 2021; Ritter, Clark, Etzioni, et al., 2011). However, more recent studies have shown that a noticeable percentage of tweets contain implicit references to named entities, referred to as implicit named entity mentions (Hosseini & Bagheri, 2020, 2021; Hosseini, Nguyen, Wu, & Bagheri, 2019; Perera, Mendes, Alex, Sheth, & Thirunarayan, 2016). Such named entity mentions lack a surface form and are only implied within the content of the tweet. For instance, in the tweet '*and that's why he is the King of Pop, Duke of Dance, Master of TheMoonwalk...but mostly the King of our Hearts for all eternity*', Michael Jackson is the main entity who is being referenced in the tweet but without being mentioned explicitly. Traditional entity linking methods and named entity taggers that are specifically developed to identify explicit entities fail to identify or link content to an appropriate entity when the named entity that is being mentioned is only implied. This is primarily due to the fact that most, if not all, entity linkers rely, to some extent, on the surface form representation of the entity to determine the entity that is being discussed.

There have been attempts to formulate the problem of implicit entity linking in tweets such as the work by Hosseini and Bagheri (2021), Perera et al. (2016) and Huang, Yuan, Zhang, and Lu (2020). The main limitation of these works is that they assume each

* Corresponding author.
E-mail addresses: hawre.hosseini@ryerson.ca (H. Hosseini), bagheri@ryerson.ca (E. Bagheri).

https://doi.org/10.1016/j.ipm.2022.102957
Received 8 October 2021; Received in revised form 13 April 2022; Accepted 23 April 2022
Available online 17 May 2022

tweet that is being processed by the implicit entity linker would always consist of an implicit entity. In other words, the assumption is that there already exists another component in the pipeline that would determine whether a tweet consists of an implicit entity or not (Hosseini, 2019). More concretely, prior work formulates the problem as follows: with an input tweet *twt*, and the type of the target implicit entity ϑ, e.g., *wikidata:Film*, implicit entity linking would retrieve an entity of type ϑ as the sought implicit entity. To the best of our knowledge, the literature has not yet explored ways through which the presence of an implicit entity within a tweet can be determined. As such, in this work, we aim at performing the task of implicit entity recognition. Formally, given an input tweet *twt* and the time *t* during which the tweet was posted, we determine if *twt* contains a mention of an implicit entity or not, in a binary classification setting. In other words, unlike the explicit entity linking task, which is concerned with whether phrases within the text can be linked to relevant entities in the knowledge graph, the task of implicit named entity recognition is focused on determining whether there are some underlying knowledge graph entities that are implied by the text but are not explicitly mentioned. We argue that this task involves deeper levels of natural language understanding than traditional named entity recognition does primarily due to the fact that explicit entity recognition relies on the surface form of explicit entities and their surrounding contextual information for identifying and disambiguating entities. However, implicit entity recognition requires the understanding of the whole grammatical and semantic structure of the content to be able to determine whether an implicit entity is being discussed or not. Therefore, our work is aware of several levels of linguistic knowledge exploiting both syntagmatic and paradigmatic perspectives. To this end, we believe that in order to be able to identify and determine the existence of implicit entities, we would need to exploit meaning in the context where it appears and as such propose to benefit from Systemic Functional Linguistics (SFL) as our theoretical framework for implicit named entity recognition.

We exploit Systemic Functional Linguistics in order to model the tweet context for the purpose of determining the presence of implicit references to entities. This theoretical approach considers meaning in language as essentially equated with the function it serves. Therefore, it aims to describe language from the perspective of its functions, rather than its structures. In computational linguistics, SFL has already been leveraged in natural language generation (Elke, Cassel, London, 1999), machine translation (Steiner, 1987) and parsing applications (O'Donnell, 1993). In the context of our work, we aim at using those linguistic components that SFL deems important for meaning formation in language and endeavor to extract patterns from such meaning formation patterns in order to determine when entity *implication* happens within text.

The choice of this framework in our work is due to manifold reasons. First, the functional approach emphasizes on the social and conceptual roles of language where it pays attention to discourse and context, hence considering language as means of communication and a semiotic system that construes meaning considering a paradigmatic point of view (Halliday & Matthiessen, 2013). Context can be significant when understanding natural language, especially within the Twitter context where users rely heavily on social context for meaning formation and communication; this is due to Twitter's social network essence (Hosseini, Nguyen, & Bagheri, 2018; Perera et al., 2016). This is not the case in the formalist approach where the focus is on the syntagmatic axis of language (O'Donnell & Bateman, 2005). Second, entities in SFL are viewed to be existent in several linguistic components and can play a key role in clauses, which is the unit of linguistic analysis in SFL. SFL generalizes patterns of referring to entities, which is convenient for computational purposes (Thompson, 2013). We hypothesize that occurrence of implicit entities inside text requires certain distinctive usage of linguistic tools and structures in isolation as well as in tandem, which we aim to analyze based on the concepts introduced in systemic functional linguistics. Finally, with the unit of analysis in SFL being clauses formed around the concept of verb processes, the components of such clauses include elements that are essential for analyzing the role of entities inside text. For instance, nominal groups – where explicit entities happen – are mainly tied together using certain verbs; analyzing these helps generalize clause level patterns, where named entities or their implicit mention may exist.

With the theoretical guidance of SFL, in this paper, we introduce and systematically classify features that are hypothesized to capture patterns of implicit entity occurrences. This allows us to develop a feature taxonomy that consists of two major classes of features namely Syntagmatic and Paradigmatic. We further exploit Syntagmatic and Paradigmatic classes in order to engineer features that are hypothesized to capture patterns resulting in implication of named entities in tweets. The first class of features, i.e., syntagmatic features, are meant to capture the local signals of implication and make use of syntagmatic linguistic tools as introduced by SFL. However, the second class of features, i.e., paradigmatic features, aim to model global characteristics of implication. Such characteristics include relationships between explicit entities within text and other textual components, drawing insights from world knowledge, i.e., structured knowledge graph data, in order to interpret patterns associated with certain entity relationships within text. With the help of these two classes of features, we build classifiers to recognize the presence of implicit mentions within a tweet.

The novel contributions of this paper to implicit entity recognition and linking literature can be enumerated as follows:

1. We formulate the problem of implicit entity recognition in tweets in a binary classification setting exploiting systemic functional linguistics as our theoretical framework for the first time;
2. We provide a systematic categorization of features and build implicit entity recognition tool that leverages syntagmatic and paradigmatic classes of features;
3. In extracting features, we curate labeled resources, e.g. process types, and publicly share such resources. Moreover, we provide a systematic method for extraction of some of the novel features based on widely accessible toolkits, such as WordNet; all of which are publicly shared with the community for further studies in this space;
4. We provide detailed feature performance and error analysis of our work to arrive at a better understanding of the task as well as for the sake of better explainability.

The remainder of this paper is structured as follows. Section 2 reviews the related literature covering computational applications of systemic functional linguistics, existing work on implicit entity recognition and linking as well as methods for named entity recognition. Section 3 then introduces how systemic function linguistics can be applied for implicit entity recognition, followed by Section 4, which proposes a collection of features based on SFL in a principled way. In Section 5, we introduce the experimental setup as well as a comparison of our work with several strong baselines. The strengths and weaknesses of our work are discussed in Section 6. The paper is finally concluded in Section 7.

2. Related work

In this section, we review works that use SFL in computational linguistics ranging from Parsing to Machine Translation and Language Generation. Furthermore, we elaborate on the related work relevant to the identification of entity implicatures.

2.1. Computational applications of SFL

The earlier applications of SFL in computational contexts relates to when SFL was used in the context of English–French Machine Translation. Booth and Firth collaborated on developing the grammar and lexicon of a machine translation system (O'Donnell & Bateman, 2005). The approach of this project was to perform word for word translation followed by some basic phrasal reorganizations. Halliday and Matthiessen (2013) later focused on developing an interlingua to which every language would be translated followed by translation into the target language. This gave way to more advanced works in this space such as the work by Wilcock (1993) who examined the benefits of using systemic functional grammar in machine translation. Bateman, Kasper, Moore, and Whitney (1990) also developed a Generalized Upper Model, which is a linguistically motivated ontology, to further build the Penman generation system used to generate sentences in English. However, using SFL for machine translation faced many problems, especially for achieving full scale systemic analyses (O'Donnell & Bateman, 2005).

Winograd was the first to develop a parsing system based on systemic functional linguistics (O'Donnell & Bateman, 2005). The system allowed human interaction with a virtual robot through a keyboard and using restricted English. Halliday's early systemic functional grammar was the basis for the development of the syntactic analyzer. Although this work was important in shaping the idea that natural language understanding was possible, it was limited in several senses, most importantly it was procedural rather than declarative. This means that it did not offer a solution that was generalizable, rather it provided a working example of a natural language understanding system. Later, inspired by Winograd's work, McCord developed a modified version of the systematic functional grammar (McCord, 1975) and subsequently implemented a parser which was slightly different than the standard SFG (McCord, 1977). Under the influence of Halliday, Kay (1985) developed a systemic grammar called Functional Unification Grammar (FUG), which is now a widely known formalism. While significantly different in several ways, FUG is clearly inspired by SFG. There were also subsequent works by Kasper (1988) and Fawcett and Tucker (1990) in the context of automatic language generation, which applied different variants of systemic functional linguistics in their work.

2.2. Implicit entity recognition and linking

The notion of implicit entities in tweets was introduced and linking of such named entities was formally defined in Perera et al. (2016), who found that a considerable percentage of tweets contain implicit mentions of entities. Perera et al. prepared and publicly shared a dataset containing tweets with implicit mentions in the two domains of movies and books. Building on the work by Hosseini et al. (2019), Perera et al. (2016) claimed that highly frequent implicit mentions of entities on Twitter is not limited to such domains as movies and books. They collected and publicly shared a dataset that contains tweets with implicit mention of entities in the domains of Person, Organization, Location, Event, Product (Device), and Work (Film and WrittenWork).

In order to identify implicit entities within tweets, Perera et al. (2016) adopt a graph-based approach. In doing so, they leverage contextual knowledge as well as exploit structured sources in order to score reachable entities from the explicit entities found in the tweet or a pool of tweets. In order to build such a graph, they use explicitly observed entities in the tweet and reachable entities derived from DBpedia's triple relations. This graph is complemented by information that is obtained from a pool of tweets that are collected in a close time frame to the original tweet. The resulting graph is used to perform the linking in two steps, namely candidate selection and candidate disambiguation. The initial phase narrows down the search space by selecting those entities that have at least one edge with matching clue nodes and tweet clues in the graph. The disambiguation phase ranks the retrieved candidates in a learning to rank environment with features extracted from the graph. Building upon this work, Hosseini et al. (2019) perform implicit entity linking in an ad-hoc retrieval framework where the input tweet is treated as a query in response to which a collection of documents are ranked. The documents represent the candidate entities to be ranked based on relevance to the input tweet. The ranking is done in the context of a Markov Random Field (MRF) framework, where several features are used as potential functions of the MRF framework. More recently, Hosseini and Bagheri (2020, 2021) proposed that implicit entity linking can be viewed as a task in the context of a learning to rank approach. To this end, they proposed a feature taxonomy that is used to train a learn to rank method. They show that such an approach is able to significantly outperform the state of the art.

A more recent approach for implicit entity linking has been proposed by Huang et al. (2020). They propose a BERT-based context representation model in order to link textual content to known entities. In order to do so, they calculate a similarity metric based on three types of contextual information that they extract from contextual embeddings. Unlike the work by Hosseini et al. (2019), Perera et al. (2016), and Hosseini and Bagheri (2021), the work by Huang et al. (2020) does not focus on Twitter content but is rather for the e-commerce domain where camouflaged product descriptions are linked to known product representations.

2.3. Named entity recognition

While the related work on implicit entities is a burgeoning discipline with only a few major recent works, the literature on named entity recognition is clearly richer and can be categorized into three types of rule-based, statistical, and hybrid approaches (Mohit, 2014). Rule-based approaches were popular at the early stages of research in this area. For such approaches, first and foremost, there was a need for rules or heuristics. Further, dictionaries (gazeteers) were required to store different named entity types. While very precise, such rule-based systems suffer from poor coverage, with their performance relying on the comprehensiveness of their rule set and lexicon. In order to incorporate more in-depth linguistic information, syntactic and statistical approaches were introduced. Such approaches also relieve the need for tedious human efforts for the curation of gazetteers and rules. Statistical models for named entity recognition, on the other hand, require labeled data for training purposes as well as a statistical model that learns the probabilistic representation. Due to the interdependence between named entities in a piece of text, this has often been modeled as a sequence labeling task, which has shown to be an effective strategy. Earliest examples of such an approach are Hidden Markov Models (Rabiner, 1989), Maximum Entropy (Berger, Della Pietra, & Della Pietra, 1996), and Conditional Random Fields (CRF) (Bikel, Miller, Schwartz, & Weischedel, 1997; Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016).

More recently, the community has focused on identifying explicit mentions of named entities using deep learning models. A dominant approach has been to use bi-directional LSTMs and CNNs (Chiu & Nichols, 2016; Jie & Lu, 2019; Zeng, Sun, Lin, & Liu, 2017). Using such deep learning models, researchers perform entity recognition and linking using context representation as well as entity representation (Ganea & Hofmann, 2017; Gupta, Singh, & Roth, 2017; Murty, Verga, Vilnis, Radovanovic, & McCallum, 2018) and transformer-based language models such as BERT (Liang et al., 2020; Souza, Nogueira, & Lotufo, 2020). For instance, Ganea and Hofmann (2017) perform entity disambiguation at the document level through learning neural representations. These authors propose a novel deep learning architecture to perform global entity disambiguation. In doing so, they learn entity embeddings through their page contents and local context of entities' hyperlink annotations. The proposed neural architecture benefits from a local attention mechanism, as the authors hypothesize that only a few context words are sufficient in disambiguating ambiguous mentions. In other works such as Gupta et al. (2017), several sources of information are encoded through deep learning architectures. Gupta and Roth jointly encode descriptions of entities, their contexts around their mentions, as well as their fine-grained types in order to disambiguate entities. They learn entity representations using such sources of information without the need for domain-specific data or manually designed features.

Similarly, Murty et al. (2018) use different sources of information, while also moving away from usually flat models by introducing hierarchical losses in their deep model. Other researchers have benefited from transformer-based language models for the task of identifying and linking explicit named entities. Liang et al. (2020) propose the BOND model, which learns to tag named entities with distant supervision. Their model is not based on specific domains or content and is useful for low resource scenarios or languages. In their two-stage approach, they use the BERT language model for refining the distant labels iteratively; this is followed by fitting the model under the teacher–student framework in the second stage. Similar to Liang et al. (2020), Souza et al. (2020) design a NER architecture using BERT and CRF for the Portuguese language.

Also relevant to our problem are works that deal with lack of abundant data. More specifically, the problem of cold start entities and entities that are missing on knowledge graphs. Such works are especially relevant as future works in implicit entity recognition will have to deal with lack of labeled data. As an example, Logeswaran et al. (2019) link entities that have not been seen before, without using in-domain labeled data. This work tries to make robust transfer to specialized, low-data domains possible. They exploit language understanding models trained on large unlabeled data, which are shown to be able to generalize to unseen instances. In a similar vein, Wu, Petroni, Josifoski, Riedel, and Zettlemoyer (2019) perform entity linking using a two-stage zero shot algorithm, which is also based on BERT. In zero-shot scenarios, machine learning models try to make predictions for classes previously unseen during training. Such works are especially relevant as creating huge datasets for implicit entity recognition needs manual labor and is not feasible to be prepared in a short amount of time. Therefore, zero-shot algorithms will be a must, where the present, relatively small datasets are leveraged to introduce algorithms that generalize well.

3. Systemic functional linguistics for implicature recognition

The investigation of entity implicature in tweets would need to involve the understanding of how implicit references occur within text. We are interested in identifying patterns that are used when implicit references to an entity are made. The identification of such patterns requires a linguistic theoretical model that focuses on meaning formation and formation of different patterns for conveying meaning. Systemic functional linguistics focuses on the systemic organization of linguistic patterns (functions) which speakers use to create meaning. Combining the notions of system and function, the idea is that language is a systemic organization of options, enabling speakers to create meaning through opting for relevant and suitable options. Different from the formalist approach, SFL does not solely focus on the structure of language, rather on the systemic choices that result in the linguistic structure. Additionally, SFL's notion of functions is tightly connected to the social uses of language, which makes it appropriate for our task. The functions of language include both the use that language serves (i.e., how and why people use language) and linguistic functions (i.e., the grammatical and semantic roles assigned to parts of language).

To this end, we exploit the categorization that SFL offers referred to as *Metafunctions*. Metafunctions correspond to the social components of language, which stipulate linguistic structures and outward forms. We believe the use of metafunctions is appropriate in our work since Twitter can be viewed as a society of speakers that speak through tweeting. The structural patterns in tweets are hypothesized to be in relation with the social context in which they are used, based on SFL. Halliday introduces three metafunctions that allow for the examination of linguistic structures and patterns as follows (Thompson, 2013):

Table 1
Metafunctions, linguistic resources, and their relevance to the task of implicit entity recognition in tweets.

Metafunction	Tool's category	Tool's sub-category	Description	Linguistic device	Relation to entities and their implicature
Textual	Cohesion Grammatical and Lexical	Reference	Involves how we point at entities in the world when we talk about them or they are involved in a situation	- Referential properties	Investigates patterns of reference where entity implicature happens.
		Reiteration	Refers to how nouns are used to replace other nouns used before.	- General nouns - Synonymy - Hypernymy	Models how implicit entities are reiterated as opposed to how explicit entities are.
		Collocation	Refers to the way words come together.	- Collocation	Models how collocation of words with explicit entities changes when the mention is implicit.
Experiential	Processes	Verb process	Pivotal for clauses and can be identified through the verbal group within clauses.	- Verb process type	Models how linguistic resources, especially verb processes, are used with named entities as participants of the process and when entity implicature happens.
	Circumstances	Adverbial groups	Adverbial phrases of a verb process are denoted circumstances.	- Circumstance type	Models how linguistic resources, especially circumstances, are used with named entities as participants of the process and when entity implicature happens.

1. *Textual metafunction*: In language usage, speakers tend to organize their messages in relation to other messages as well as with respect to the wider context in which the linguistic utterance occurs;
2. *Experiential metafunction*: Language is used by speakers to talk about their experiences of the world in order to describe events and situations and entities playing a role in them;
3. *Interpersonal metafunction*: Speakers also use language to communicate with other people for different purposes, including building relationships, expressing perspectives on different things, among others.

In our work, we adopt the first two metafunctions as the basis for investigating meaning creation and comprehension. The reason interpersonal metafunction is not considered is because it would only be meaningful within the context of dialog and not necessarily in the context of individual tweets. In what follows, we elaborate on textual and experiential metafunctions within SFL. Table 1 provides an overview of the two metafunctions and how they relate to the task of implicit entity recognition.

3.1. Textual metafunction

As expressed by Halliday and Matthiessen (2013), the purpose of the textual metafunction is to allow the speaker to organize the message they would like to convey and differentiate between language in the abstract and language in use; therefore, make the message they intend to communicate relevant to their context. We focus on two major components in textual metafunction in the following:

3.1.1. Reference

Reference involves how we point at entities in the world when we talk about them or they are involved in a situation. The usage of reference serves the broad function of showing how the message fits into its context; exophoric reference links the language to the external context, while endophoric reference signals how the message fits specifically into its textual context (the 'co-text'). It is the latter – reference as cohesion – which we will focus on in our work. Most cohesive references are anaphoric ('pointing backwards'): the meaning that is being repeated has already been mentioned earlier in the text. Less often, reference may be cataphoric ('pointing forwards'): this signals that the meaning of the reference item will not be specified until further on in the text. References are key in our implicit entity recognition task since we would like to investigate patterns of reference to implicit entities where entity implicature happens.

3.1.2. Lexical cohesion

Lexical cohesion refers to the way words are chosen to link elements of a text in relation to each other. The most related lexical cohesion devices to our work are reiteration and collocation. Here, we elaborate on each of these two devices and how they are related to the recognition of implicit entities. Within text and dialog, nouns are sometimes used to replace other nouns, so called *reiterated*. *Reiteration* is accomplished through different schemas which are categorized into several different types in the context of SFL. For instance, the 'general nouns' schema refers to a set of nouns that have a generalized reference in the major classes of nouns, e.g., people, person, stuff, thing, movie, place, and so forth. For the purpose of this study, such general nouns can provide signals that entities are replaced by such nouns and therefore there is a missing entity being implied. Other schemas for capturing reiteration

include synonymy and hypernymy. Another lexical cohesion device is *collocation*, which refers to the way words come together. The co-occurrence of words entails cohesion or lack of cohesion in the sense that items within a cohesive text are homogeneous with respect to the context and the formed meaning. In the context of our work, collocation is an important tool since it models how entities are collocated with other words. More concretely, lack of an entity which is normally collocated with a set of other words within text can be a signal for the implication of an entity.

3.2. Experiential metafunction

From an experiential metafunction point of view, language is viewed as a set of resources for the purpose of referring to world entities, the relationship between these entities, and how these entities act. Therefore, named entities within text are involved when relevant linguistic resources are used. We are interested in identifying the patterns that are formed with regards to usage of such resources when entity implicature happens. The experiential metafunction has two major linguistic tools, namely *processes* and *circumstances*. Processes play an important role in clauses and can be identified through verbal groups. On the other hand, circumstances are realized or expressed through adverbial groups or prepositional phrases.

With regards to processes, the focus can be on process type as well as on the relationship between the process and participants. Based on SFL, process types have the following categories: Material, Mental, Relational, Verbal, Behavioral and Existential. Such process types are categorized based on the events in our world and inner experiences we have from them according to our interpretation. These interpretations are formed through connections or generalizations between what we have experienced before and what we are experiencing now. For instance, 'mental' processes represent the inner world such as thinking, understanding and imagining as the processes happening in our mind. As another example, 'relational' processes connect our inner world (our mind) and the outer world. For instance, when a participant in the process attributes a property to another participant while the two participants are not independent of each other. In the context of our work, analyzing processes will help us in recognizing entity implicature since the processes represent how the entities happen in the inner world or the outer world as well as the relationship between entities as participants of the process.

Similar to processes, circumstances are categorized into different types. There are nine types of circumstance, namely Location, Extent, Manner, Cause, Contingency, Accompaniment, Role, Matter and Angle. The categorization of circumstances in SFL is based on the specific type of meaning they form. In the context of our task, the consideration of circumstances is important since they introduce entities as indirect participants, hence giving further information about the entity. For example, in the tweet '*Jurassic Park 3 is the worst thing ever made by humanity*.', the circumstance is 'by humanity'. This circumstance is of type *Means* and subtype of *Manner*. The additional information conveyed through this circumstance can provide us with clues about the entity and its properties.

4. Proposed framework

4.1. Hypothesis development

We base our arguments on the hypothesis that implication of entity mentions follows certain patterns in linguistic utterances. These utterances are argued to be natural language instances in terms of their structure, similar to instances without implicit entity references. Therefore, we suggest that distinguishing between implicit and explicit tweets is not necessarily a problem of investigating well-structuredness of language; rather, one of identifying the patterns that are formed through a combination of syntagmatic and paradigmatic features in order to imply a named entity. Based on these, we hypothesize that implicit tweets contain generalizable structural and semantic features which are distinctive from explicit tweets that can be exploited within the context of recognizing implicit entities in tweets.

As mentioned earlier, SFL offers the classification of syntagmatic and paradigmatic elements for language components that together form meaning. We hypothesize that SFL offers tools that are useful for the recognition of implicit entities; therefore, we propose that implicit mentions of entities in tweets can be recognized through leveraging SFL based identification of significant linguistic elements in meaning formation.

4.2. SFL-based feature engineering

According to SFL, we propose that features can be designed for the task of implicit entity recognition based on two main categories, namely Syntagmatic and Paradigmatic. Syntagmatic and paradigmatic approaches in linguistics try to capture how language works from different points of view. The syntagmatic approach studies the way linguistic units can be combined in order to develop more complex units and bear syntagmatic properties. The paradigmatic approach examines the relationship between linguistic units where one linguistic unit in one position can be replaced with another linguistic unit in the same position. We exploit this classification for designing features since we argue that the recognition of implicit entities requires analysis on both perspectives. From the syntagmatic viewpoint, implication and more specifically implicit mentions can happen due to a certain set of word combinations with certain attributes such as syntactic dependencies and coreferential properties, among others. Additionally, from a paradigmatic point of view, the position of an entity might be filled by other semantically related linguistic components. Analyzing such replacements can help us recognize when an implicit mention happens. Fig. 1 provides an overview of the proposed feature taxonomy based on the syntagmatic and paradigmatic approaches. Please note that while elaborating on the features in the following, example tweets are referred to in the text by the unique code they have been assigned in Tables 2, 3, and 5.

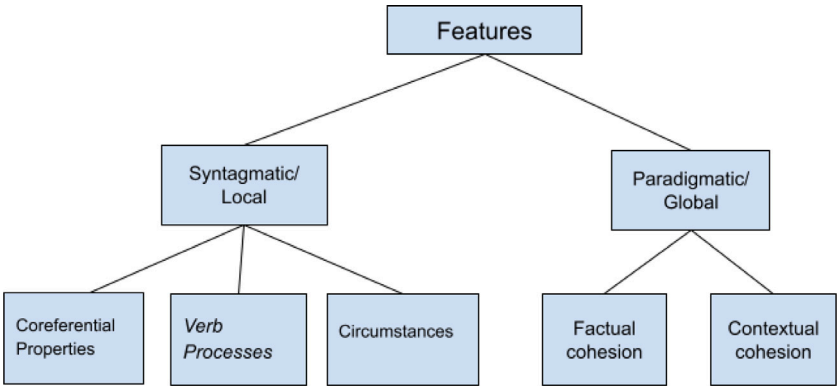

Fig. 1. An overview of the proposed feature taxonomy.

Table 2
Extracted values for features based on coreferential properties. The column code is for easier reference to the tweet inside text.

Target	Code	Tweet example	SR1	SR2	SR3	SR4	SR5	SR6	SR7
Implicit	A	*'he should've won an Oscar for Titanic and The Wolf of Wall Street'*	*T*	*F*	*F*	*he*	*NoType*	*NoType*	*he*
Implicit	B	*'He will forever be the King of Pop. I wish I could've seen him live once. Now the only thing I can do is blast his music so my neighbors can know what real music used to be lol'*	*T*	*T*	*F*	*He Him his*	*wikidata:Person*	*NoType*	*He, I, him, my*
Implicit	C	*'He just did Baywatch, doing Jumanji, Hobbs spinoff, and Fast 9 in 2020....dude is set'*	*T*	*F*	*F*	*He*	*NoType*	*NoType*	*he*
Explicit	D	*Ellen Pompeo just gave the best interview I've read since Jen Aniston dished on her divorce from that idiot Brad Pitt*	*F*	*F*	*F*	-	*NoType*	*NoType*	*her*

4.2.1. Syntagmatic features

The purpose of syntagmatic features in our work is to capture the patterns that are formed when named entities are mentioned implicitly.

Coreferential Properties. Understanding text is often achieved through its cohesion. In the SFL approach, especially within its textual metafunction, the issue of cohesion is important since cohesion is the linguistic device for expressing coherent meaning. There are several characteristics and tools for cohesion as outlined earlier in Section 3.1. We believe that lexical cohesion and reference can be influential for the purpose of implicit entity recognition. The latter, i.e., reference which is mainly accomplished through Personal Reference, is essential when it comes to the study of named entities. The reason is that Personal Pronouns are used in order to refer to named entities. They include pronouns and possessives. Such references need to be investigated in order to find patterns signifying reference to implicit mentions of entities. We endeavor to capture such patterns through engineering features that investigate several types of references made using pronouns. We enumerate such patterns from within the context of SFL as follows, and show the values of each feature for several tweets in Table 2.

One of the signs of the presence of an implicit entity is when there is no coreferent for a pronoun. This is an indication that the entity is being implied. We defined two features for this pattern as follows:

Syntagmatic Reference Feature 1 (SR1). The first feature is a boolean feature that is true if the following condition holds:

$$r_c^T \in R_c^T, and$$

$$r_c^T = (p \rightarrow \varphi).$$

where R_c^T stands for the set of tuples of coreferential relations r_c^T inside text T between personal pronoun p and another component. In other words, the feature determines whether there is a personal pronoun whose coreferent does not exist within the text. In the example tweet A, there is only one coreferential relation: $(p \rightarrow \varphi)$, where φ is not to be found inside the tweet making the feature value *True*. Compare this to the other example tweet D in which the coreferent in the coreferential relation *(her → Jen Aniston)* is to be found within the text, i.e. Jen Aniston, making the feature value *False*.

Syntagmatic Reference Feature 2 (SR2). Similar to SR1, the second feature is also a boolean feature. This feature leverages anchors, which are clickable chunks of text on Wikipedia that are linked to other Wikipedia pages. For instance, the phrase '44th president of the United States of America' is an anchor which links to *Barack Obama*'s page. This feature's value is determined based on the following condition:

$$r_c^T \in R_c^T, and$$

$$r_c^T = (p \rightarrow NG), and$$

$$NG = a, and$$

$$e^a \notin E^T.$$

where NG stands for a nominal group, e represents an explicit named entity, a is an entity anchor, e^a is the explicit entity that is linked by anchor a, and E^T is the set of explicit entities that are inside text T. In other words, this feature checks if there is a personal pronoun in the text whose resolved coreferent is an entity anchor whose corresponding explicit entity is not found within the text. Take tweet B for an example. In this tweet, the coreferenced phrase 'King of Pop' is identified as a Wikipedia anchor that refers to entity *Michael Jackson* and this entity is not to be found inside the tweet making the value of the feature *True*.

Another sign of an implicit entity within the context of SFL is that the reference is to another nominal group which itself is not an anchor, but is in an inheritance relationship with an anchor. We define an additional feature based on this pattern as follows:

Syntagmatic Reference Feature 3 (SR3). Based on the second pattern, we define a boolean feature that explores the relation between the reference and a nominal group as follows:

$$r_c^T \in R_c^T, and$$

$$r_c^T = (p \rightarrow NG), and$$

$$NG \neq a, and$$

$$r_{is-a}^T = (NG \rightarrow a), and$$

$$e^a \notin E^T.$$

Take the following tweet as an example: 'This guy is the King of Pop for a reason. I just love him.' Here, the coreferential relation is between 'him' and 'This guy', while 'This guy' is in an *is_a* relationship with 'King of Pop', which is an identified anchor phrase referring to Michael Jackson.

Syntagmatic Reference Features 4, 5, 6 and 7 (SR4, SR5, SR6, SR7). Furthermore, understanding the patterns formed by collocation of types of pronouns as well as classes of entities referred by anchors can help determine whether an implicit entity is present in the text or not. More specifically, we aim at identifying reinforcing effects that different components of reference patterns can have. Therefore, based on the previous features, we propose features to extract additional information in relation to SRs 1–3, as elaborated on in the following. Feature SR4 identifies the personal pronoun p in SR1. The feature value would equal to a list of all the pronouns whose coreferents are not resolved. Therefore, for instance, the feature value in the example tweet A would be *'he'*. SR5 and SR6 determine the DBpedia type rdf:type of the explicit entity that is linked by anchor a in SRs 2–3. In this work, we consider our coarse-grained classes as the type of entity. Therefore, we map any type returned by DBpedia to one of these types, even if the value returned from the query is a fine-grained type. In the examples of these two tweets, the anchor phrase is linked to *Michael Jackson* whose type is dbo:Person. Finally, SR7 collects the set of pronouns that are observed in the text. This way, the value of this feature with the example tweet B as input would be *'He, Him, his'*. These features would allow us to go beyond the reference and the coreferent and actually consider the values and types of the references. The reason for this is that certain references and types are more likely to be indicators for the presence of an implicit entity.

Verb Processes. In the context of textual metafunctions, we design additional features based on the types of verb processes in combination with explicit entity types present in the verb arguments. The verb processes and their significance to the SFL are elaborated on in Section 3.2., where experiential metafunction is described. The intuition is that certain implicit mentions of entities happen when particular explicit entities participate in verb processes. For instance, to implicitly mention an actor in the context of a recently released movie, the presence of the explicit entity of that movie and a process (verb) that is participated in by an explicit entity of type Person (actor) is required. On this basis, we propose two features that investigate the verb processes and their participants. As described earlier, based on SFL, process types have the following categories: Material, Mental, Relational, Verbal, Behavioral and Existential. We manually create a list of verb process types for different verbs to help us in the extraction of the features. The two features are as follows:

Syntagmatic Verb Process Feature 1 (SVP1). The types of verb processes employed within the text may be indicators for the presence of implicit entities, as such, this feature captures the types of verb processes that are observed in the text. To extract this feature, we syntactically parse the input tweets in order to identify the verbs that exist within the input and extract the process type using our manually created verb process type set. In the example tweet A, we identify the verb 'win' in the text and extract its process type, i.e., *mental*.

Syntagmatic Verb Process Feature 2 (SVP2). To further augment SVP1 with additional semantic information, we employ WordNet-based sense information to determine the sense of the very process used in the text in this feature. In doing so, we query WordNet to find all of the verbs' lexnames of the verb senses to form a set of such word senses. During WordNet development, each synset has been assigned to a lexicographer file. As generalizations, these lexicographer files (lexnames) group different synsets together logically as well as based on syntactic category.

Table 3
Extracted values for features based on verb processes and circumstances properties. The column code is for easier reference to the tweet inside text.

Target	Code	Tweet Example	SVP1	SVP2	SVP3				SC1	SC2
Implicit	*A*	*'he should've won an Oscar for Titanic and The Wolf of Wall Street'*	*mental*	*verb.competition*	*F*	*F*	*F*	*F*	*cause*	*reason*
Implicit	*B*	*'He will forever be the King of Pop. I wish I could've seen him live once. Now the only thing I can do is blast his music so my neighbors can know what real music used to be lol'*	*v1:relational v2:mental*	*v1:be:verb.stative v2:see:verb.perception*	*T*	*T*	*music*	*F*	*extent, location, cause*	*Frequency, time, reason*
Implicit	*C*	*'He just did Baywatch, doing Jumanji, Hobbs spinoff, and Fast 9 in 2020....dude is set'*	*material*	*verb.creation*	*F*	*F*	*F*	*F*	*location*	*time*
Explicit	*D*	*Ellen Pompeo just gave the best interview I've read since Jen Aniston dished on her divorce from that idiot Brad Pitt*	*material*	*verb.possession*	*T*	*T*	*divorce*	*T*	*cause*	*reason*

Given the fact that entities can be interrelated in the real world through different relationship types, it is possible to identify attributes related to entities which are implicitly mentioned. For instance, an explicit entity can be related to another implicitly mentioned entity through such relationships as spouse, boss, partisanship, co-founder, among others. On the other hand, one of the main methods of referring to a related entity in natural language is through use of possessive phrases. Therefore, we design the following feature in order to capture such relationships as well as patterns formed through such uses of natural language.

Syntagmatic Verb Process Feature 3 (SVP3). This feature focuses on whether reference is being made to a related entity in the text through possessive phrases. To this end, the feature will identify whether there is a possessive noun chunk participant of a verb process in the text and what are its associated category related keywords and whether the possessor is an explicit entity or a pronoun referring to an explicit entity or not. Take tweet *B* for an example. There is a possessive structure 'his music' making the first feature value *True*. The possessed is checked to see if it is a category related keyword. In this case, 'music' is the possessed noun and it is related to the category Work (which includes categories Film and Book in our dataset), making the next feature value True and the third feature value equivalent to *Work*. These category-related keywords are adopted from Hosseini et al. (2019). These keywords, as authors report, were selected by the annotators who were assigned the task of labeling the gold standard dataset. These lists contains keywords which were agreed on by all the annotators. For the fourth feature in this group, we look to see if the possessor is an explicit entity or a pronoun referring to an explicit entity. In this example, 'his' is neither an explicit entity nor a pronoun referring to an explicit entity. For the sake of comparison, observe tweet *D* as an example. Here, the pronoun 'her' in the possessive chunk 'her divorce' refers to 'Jen Aniston' which is an explicit entity, making the value for the last feature *True*.

Similar to features related to coreferential properties, we present the values for the features defined based on verb processes in Table 3 to show the values for these features are computed.

Circumstances. Circumstance is one of the elements of experiential metafunction, which is often a prepositional phrase or adverbial group and is often optional and in some cases obligatory. According to Thompson (2013), circumstances reflect "background" function in the clause. The importance of circumstance is where an entity appears to give us information about another entity. Halliday and Matthiessen (2013) have proposed a classification for circumstance types. We adopt circumstances and their types to define features as they can be strong indicators as whether the circumstances are providing clarifying information about an entity that is not present in the text, hence pointing to the presence of an implicit entity. For a detailed discussion of circumstance within SFL, please refer to Section 3.2., which discusses experiential metafunction.

The associated extraction procedures are outlined in Table 4. In extraction of circumstance features, the input tweet needs to be parsed in order to recognize the phrases that qualify to be circumstances. Afterwards, the circumstance typing procedures are leveraged in order to identify the circumstance's coarse-grained and fine-grained classes. In doing so, the WordNet is used for this purpose.

Syntagmatic Circumstance Feature 1 (SC1). Based on circumstance types defined by Halliday and Matthiessen (2013), we hypothesize that there are circumstance types that have a higher likelihood of appearing in contexts where implicit entities are observed. As such, this first feature reflects the coarse-grained type of circumstances as shown in Table 4.

Syntagmatic Circumstance Feature 2 (SC2). We further hypothesize that finer-grained sub-type information within circumstance types, such as time and place finer-grained subtypes for the location circumstance type, are discriminative features for determining whether an entity is implied within text or not. Detailed finer-grained sub-types without a referent can indicate the presence of an implicit entity; therefore, this feature captures finer-grained circumstance sub-types.

We note that Table 3 presents SCs 1–2 feature values extracted for some sample tweets to show how the values for these features are computed. As an example, consider tweet *C*. Here, the identified circumstance phrase is 'in 2020'. Using the extraction procedure, it is apparent that this circumstance is qualified for one of type *Location* and sub-type of *Time*.

Table 4
Extraction procedures for coarse- and fine-grained types of circumstances.

Type	Sub-type	Instrument
Location	1. Time	- at, in, on, to, until, till, towards, into, from, since, during, before, after + [WN-noun.time NG head] - adverb of time: today, yesterday, tomorrow, now, then + [WN-noun.time NG head]
	2. Place	- at, in, on, by, near, to, towards, into, onto, (away) from, out of, off, behind, in front of, above, below, under, alongside + [WN-noun.location NG head] - adverb of place: abroad, overseas, home, upstairs, downstairs, inside, outside, out, up, down, behind, left, right, straight, there, here
Extent	1. Distance	(For) + [measured nominal group, e.g. WN-noun.quantity NG head]
	2. Duration	(For) + [measured nominal group, e.g. WN-noun.time NG head]
	3. Frequency	Measured nominal group of type frequency, e.g. every, once, twice, etc.
Manner	1. Means	[by, through, with, by means of, out of, from] + [material noun, e.g. WN-noun.artifact NG head]
	2. Quality	- fast, well, together, jointly, separately, respectively
	3. Comparison	- WN-noun.resemblance, e.g. like, unlike, similar to, different from
	4. Degree	- to a [high, low] + [WN-noun.quantity NG head, e.g. degree, extent] - adverbs of degree: much, greatly, considerably, deeply, completely
Cause	1. Reason	Because, because of, as a result of, thanks to, due to, for, of, out of, through
	2. Purpose	for, for the purpose of, for the sake of, in the hope of
	3. Behalf	on behalf of
Contingency	1. Condition	in case of, in the event of, if, when, while, once
	2. Default	in default of, in the absence of, short of, without
	3. Concession	despite, in spite of
Accompaniment	1. Comitative	[With, without, alongside] + [WN-noun.person, WN-noun.animal]
	2. Additive	[as well as, besides, instead of, in addition to] + [WN-noun.artifact NG head, e.g. gloves]
Role	1. Guise	[as, by way of, in the role/shape/guise/form of]
	2. Product	[into, to] + [WN-noun.artifact/WN-noun.substance/WN-noun.cognition/WN-noun.phenomenon NG head]
Matter	Matter	–
Angle	1. Source	[according to, in the words of] + [WN-noun.person/WN-noun.location NG head]
	2. Viewpoint	[to, in the view/opinion of, from the standpoint of] + [WN-noun.cognition/WN-noun.location NG heads]

4.2.2. Paradigmatic features

Given the fact that identifying and understanding implicit entity mentions are dependent on the context they are being used, we propose that the adoption of paradigmatic knowledge can be impactful in the context of implicit entity recognition. A paradigmatic analysis can help generalize the concepts around entity mentions and between different entities mentions. This generalization will in turn identify the patterns through which implicit entities are referred to in the context of tweets. We propose class of paradigmatic features in order to capture contextual information.

Factual Cohesion. Cohesion concepts are employed to understand the underlying model of meaning formation. Such cohesion related concepts and the relevant linguistic tools within the SFL are described in Section 3.1.2., where lexical cohesion is described. For the purpose of implicit entity recognition, for example, a set of linguistic components which may help identify implicit mentions and their types is word senses based on an external knowledge base such as WordNet, which captures such lexical cohesion tools as hypernymy. Additionally, it is hypothesized that entities of certain types participate in certain verb processes and collocate with certain word senses. As seen in the example tweet '*It's time to tear down that statue in New York*', the verb *tear down* can be used for an object of type *Structure*, i.e., a word with hypernym structure, among others, and cannot normally be used for objects of type *Person*. These, in tandem with the relation of entities can identify implicit entity mentions.

From an interrelated knowledge graph perspective, named entities are essentially related to each other forming a factual network of relations. Such relationships are reflected in text where two named entities are mentioned. Therefore, we are interested in features that can capture entity relationships. This is due to the fact that sometimes entities are implicitly mentioned through their relationships with other explicit entities inside the text. In the example tweet, and when considering information from an external knowledge base such as DBpedia, one can see that *The Statue of Liberty* forms the relationship dbp:locmapin with *New York*. Hence, while *The Statue of Liberty* is not mentioned in the tweet, its relation to the explicitly mentioned *New York* entity and the fact that the term statue appears in the tweet would allow us to identify an implicit entity in the tweet. In order to be able to define factual cohesion features, we assume access to a knowledge graph where both explicit and implicit entities are contextualized.

One of the indicators of an implicit entity in text can be cases when one would explicitly observe the subject (object) and the predicate of a relationship that is present on the knowledge graph but not observed in the corresponding object (subject) in the text. This could indicate that the text is implying the missing element of the triple and hence it can be an indicator for an implicit entity. On this basis, facts in triple forms from a knowledge graph would allow us to check for factual cohesion.

Table 5
Extracted values of factual cohesion features for sample tweets. The column code is for easier reference to the tweet inside text.

Target	Code	Tweet Example	FC1	FC2	FC3	FC4	FC5	FC6
Implicit	A	*'he should've won an Oscar for Titanic and The Wolf of Wall Street'*	*T*	*F*	*F*	*T*	*F*	*wikidata:Film wikidata:Award*
Implicit	B	*'He will forever be the King of Pop. I wish I could've seen him live once. Now the only thing I can do is blast his music so my neighbors can know what real music used to be lol'*	*F*	*F*	*F*	*T*	*T*	*NoEntity*
Implicit	C	*'He just did Baywatch, doing Jumanji, Hobbs spinoff, and Fast 9 in 2020....dude is set'*	*T*	*F*	*F*	*T*	*F*	*wikidata:Film*
Implicit	E	*'Deepika Padukone & SRK in a movie directed by Farah Khan. Praise this news.'*	*T*	*T*	*F*	*T*	*F*	*wikidata:Person*
Explicit	F	*Machu Picchu is an unique place*	*F*	*F*	*F*	*F*	*F*	*wikidata:Location*

Factual Cohesion Feature 1 (FC1). The first feature in this class is a boolean feature that checks whether a relation triple R^T inside text T aligns with a relation R^{KG} on the knowledge graph KG with two named entities e_1 and e_2 where $R^{KG} = (e_1, r, e_2)$ and $R^T = (-, r, e_2)$ or $R^T = (e_1, r, -)$, where an entity is missing in either the subject or the object in R^T. As an example, consider tweet *A*. Here, the triple between 'he', 'should've won' and 'Oscar' is identified using our open information extraction module. After extraction of triples, if a phrase is a surface form of an explicit entity, it is replaced with its corresponding entity. Then, the triples are mapped to DBpedia triples. In this example, we recognize that many entities are connected with Academy Awards, that is the Oscars, all of which are missing in the input tweet.

It is possible to further refine FC1 to cover cases where the specific subject or object in R^{KG} are omitted in R^T but instead their hypernym is present. For instance, instead of saying '*The Godfather* was a great watch', the text would say '*The Film* was a great watch.'

Factual Cohesion Feature 2 (FC2). This feature specialize FC1 by exploring whether a hypernym of the subject or object of e_1 or e_2 in $R^{KG} = (e_1, r, e_2)$ is present in R^T and either e_1 or e_2 is omitted from the text. Take tweet *E* as an example. In this tweet, the triple between 'a movie', 'directed by', and 'Farah Khan' is identified using our information extraction module. Once mapped to the DBpedia triples, we realize that there are triples that correspond to this triple if we map their subjects (which are movies) to their categories.

We further generalize FC2 to allow for both subject and object to be mentioned implicitly within the text. For cases where both subject and object of R^{KG} are missing from the text but instead there exists another relation $R^T = (e_1, r, e_2)$ in the text whose subject and object are both hypernyms for the subject and objects of R^{KG}.

Factual Cohesion Feature 3 (FC3). It is possible for both subject and object of a factual knowledge graph triple to be implied in the text where both subject and object of R^{KG} are missing from the text but instead there exists another relation $R^T = (e_1, r, e_2)$ in the text whose subject and object are both hypernyms for the subject and objects of R^{KG}. As an example, observe the tweet 'This guy is the King of Pop for a reason. I just love him.' where 'guy' (as mapped to Person) is a hypernym of the explicit entity of type Person and 'King' is also a hypernym for explicit entities of type Person. Therefore, the feature value for such a tweet would be *True*.

In FC2, the triples are mapped to KG triples. Mapping relationships is a hard task to do. In order to reduce sparsity of FC3, we relieve the alignment condition for the triple, meaning that the triple's relation is not considered for the match, but only subject and object of the triple.

Factual Cohesion Feature 4 (FC4). This feature specializes FC2 by exploring whether a hypernym of the subject or object of e_1 or e_2 in $R^{KG} = (e_1, r, e_2)$ is present in R^T and either e_1 or e_2 is omitted from the text, with the difference that the relation r of triple $R^T = (e_1, r, e_2)$ inside text does not need to align with a relation R^{KG} on the knowledge graph KG. In example tweet *E*, the triple between 'a movie', 'directed by', and 'Farah Khan' is identified. When mapping this triple to KG triples, we match it with any triple whose subject (or object) is Farah Khan and their object (or subject) is a movie, regardless of the relation being that of *dbo:director*.

The assumption of the previous four features is that the explicit entities observed in the text are accurately extracted by an explicit entity linking system. However, in reality, there are many cases where an explicit entity linker will miss accurately identifying all of the explicitly mentioned entities. We further define an additional feature to cover such cases as follows:

Factual Cohesion Feature 5 (FC5). This feature will identify cases where an anchor a in the text has partial or complete textual overlap with the description of an entity e^a in the knowledge graph but e^a is not present in the tweet's set of explicit entities E^T, i.e., $e^a \notin E^T$. As an example, observe that the anchor phrase 'King of Pop' in tweet *B* is referring to Michael Jackson, while Michael Jackson is not in the linked entities of the tweet.

Finally, we hypothesize that certain types of entities have a higher likelihood of being implied in text compared to other entities. However, given implied entities are not observed in text, we propose a feature to consider the association between explicitly observed entities in text and implied entities and use this as an indication of an implied entity. In other words, based on the likelihood of an explicit entity being associated with an implied entity, we would infer the presence of an implied entity if the associated explicit entity is observed.

Factual Cohesion Feature 6 (FC6). This feature captures types of all of the explicitly mentioned entities within the text in order to allow for identifying associations between explicit and implicit entities. As an example, note how the feature value for the tweet 'he should've won an Oscar for Titanic and The Wolf of Wall Street' is *wikidata:Award, wikidata:Film* since the explicit entities Oscars and the Wolf of Wall Street are linked to their corresponding KG entries.

Table 6
An overview of the implicit entity recognition and linking gold standard dataset, introduced by Hosseini et al. (2019).

Type	Implicit	Explicit	No Entity (NE)
Count	1,345	2,483	3,842
Average explicit entity per tweet	2.53	2.68	0
Average token per tweet	26.16	21.96	16.6

We note that the values for the features proposed under factual cohesion are reported for sample tweets in Table 5.

Contextual Cohesion. Frequency of entity mentions and how they are referred to on social platforms is dynamic and context dependent. For instance, a criticism towards a specific senator can be made by various groups at different times or a scandalous divorce report can happen for different celebrities. For this reason, we believe cohesion of information presented in the text can also be viewed from within the context of the broader information presented on the social network. As such, while in factual cohesion, we viewed cohesion from the perspective of how information presented in the text aligns with factual information presented on the knowledge graph, here in contextual cohesion, we explore how information presented in the text aligns with information that is being shared on the social network with the specific time frame when the text (tweet) was published.

In order to define contextual cohesion features, we adopt the same strategy as those for factual cohesion except that instead of considering factual relations from the knowledge graph, we will rely on explicitly observed relations between explicit entities on the social network. To identify such relations, we pool tweets from the same time period as when the tweet under consideration was published. Afterwards, we extract information with respect to how explicit entities and hypernyms appear in relationship with each other inside the pooled tweets. Once these relation triples are extracted we, consider them to be the set of contextualized relations that will be used for measuring contextual cohesion features.

We adopt the same set of FCs1-5 for the contextual cohesion features. To avoid repetition, we only represent how the first feature FC1 is adopted here and the rest of the features are defined similarly.

Contextual Cohesion Feature 1 (CC1). This boolean feature checks whether a relation triple R^T inside text T aligns with a relation R^{pooled} obtained from the set of pooled tweets published at the same time as T such that $R^{pooled} = (e_1, r, e_2)$ and $R^T = (-, r, e_2)$ or $R^T = (e_1, r, -)$, where an entity is missing in either the subject or the object of R^T. In tweet A, for instance, the triple between 'he', 'should've won' and 'Oscar' is identified using our open information extraction module. After extraction of triples, if a phrase is a surface form of an explicit entity, it is replaced with its corresponding entity. Then, the triples are mapped to triples extracted from the set of pooled tweets. We check if such a relation has been observed on the social network or not to determine the feature value.

The major differentiating factor between factual cohesion and contextual cohesion features relates to the set of relations that are adopted. In factual cohesion, all relations in the knowledge graph are considered to identify an implied entity, whereas in contextual cohesion, only relations obtained from the social network as contextual information are considered which are related to the explicitly observed entities of a set of pooled tweets that appeared at the same as the tweet under consideration. The reason behind this is that the likelihood of a relation from the knowledge graph being related to a tweet may be related to the set of explicit entities that are observed in other related tweets in the same time period. For instance, a tweet such as 'what a majestic event' at the time when all other tweets published at the same time are referring to the opening ceremony of the Olympics, is also likely implying the same event.

We note that the other factual cohesion features are similarly defined for contextual cohesion as well.

5. Experiments

The main objective of this paper is to address the problem of implicit entity recognition in tweets by proposing a systematically classified set of features that could be appropriate for that task, all in a framework inspired by systemic functional linguistics. We evaluate our proposed approach and compare it with baselines both quantitatively and qualitatively. Further, we evaluate our hypotheses presented earlier based on different feature set ablations. Lastly, we present a detailed error analysis of errors made by our approach in order to identify the weaknesses of our proposed features as well as areas that can be adopted for future work.

5.1. Dataset

For our experiments, we adopt the gold standard dataset proposed in Hosseini et al. (2019), which is specifically collected and tagged for the task of implicit entity recognition and linking in tweets. This dataset, as described in Hosseini et al. (2019), resembles traditional NERC tasks datasets with a two-level hierarchy of fine- and coarse-grained hierarchy. The dataset consists of 6 coarse-grained entity types of *Person, Organization, Location, Product/Device, Event,* and *Work*. The classes contained are inspired by classes of entities based on the DBpedia taxonomy. Tweets in this dataset consists of three categories, namely those containing implicit mentions of an entity, denoted as Implicit tweets; tweets including explicit entity mentions with no implicit mention of entities, denoted as Explicit tweets; and tweets without either of the aforementioned types of mentions, denoted as No Entity (NE). Table 6 and Fig. 2 summarize the statistics of this dataset.

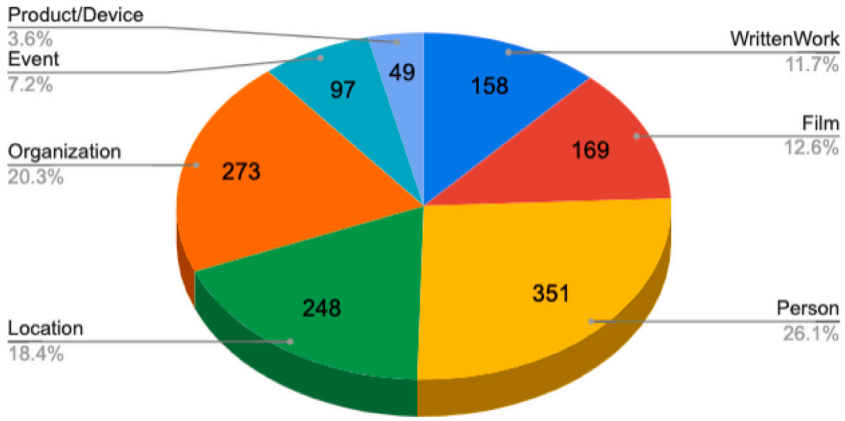

Fig. 2. Distribution of tweets per coarse-grained class in the gold dataset.

Table 7
Performance of our proposed approach for implicit entity recognition based on precision, recall, and F1 score.

Recognition	Precision				Recall				F1			
	fastText	StarSpace	BERT-based	Our approach	fastText	StarSpace	BERT-based	Our approach	fastText	StarSpace	BERT-based	Our approach
Overall	58.73	55.32	68.03	**72.71**	45.12	57.63	68.83	**73.63**	46.19	56.13	66.47	**72.77**
Positive class only	0.5	35.93	56.5	**62.18**	17.39	25	35.93	**50.92**	25.8	29.48	42.28	**55.96**

5.2. Experimental setup

Here, we explain how the manual collection and labeling of the pooled tweets required in the contextual cohesion features was performed according to Hosseini et al. (2019). For each tweet, we pooled tweets using the Twitter API for a period of two weeks surrounding the time the tweet under consideration was published. In order to extract explicit entities in tweets, we adopted the TagMe API, which has been shown to have reasonable performance in the context of Twitter (Ferragina & Scaiella, 2010). Furthermore and where required, we employed the SpaCy NLP framework[1] for dependency parsing, chunking, and coreference resolution of textual content. Also, for relation extraction, we use StanfordNLP language processing toolkit.[2] We also use NLTK[3] for extraction of word senses. For anchor related features, we build a mapping from entity anchors to entities using Wikipedia dumps from December 20, 2018. We note that all the results reported in this paper are based on a five-fold cross validation strategy trained using the extra-trees classifier, which is an extension of the widely used random forest algorithm. There are several reasons why we opted for extra-trees algorithm. First, we aimed for an algorithm that would enable us to extract feature utilities based on known metrics. A tree-based ensemble algorithm would satisfy this by providing the opportunity to measure the Gini score that can be used for identifying feature utilities. Second, the extra-trees classifier is known to be a suitable choice for scenarios where high predictive power is desired without risking overfitting (Geurts, Ernst, & Wehenkel, 2006; Geurts & Louppe, 2011). Considering the fact some of our feature values are noisy with outliers, we opted for this choice.

The code, data and results for our work in this paper are accessible on Github.[4]

5.3. Baselines and evaluation metrics

In order to evaluate the performance of our approach and compare with the baselines, we adopt widely used criteria and metrics used for binary classification problems. In such an environment, an input text, here tweet, is classified as either containing implicit mentions, denoting it an implicit tweet, or not, hence denoting it explicit. The main evaluation metrics used are Precision, Recall and F1 score. In order to establish baselines, we opted for three recent neural text classification techniques, namely *fastText* (Joulin, Grave, Bojanowski, & Mikolov, 2017) and *StarSpace* (Wu et al., 2018) and a *BERT-based* classifier (Minaee et al., 2021). As suggested by the authors of the baseline methods, the following specifications were adopted for the hyperparameters of the neural models: the word embedding dimension is set at 200, learning rate of 0.1, word Ngrams of 2, and 25 epochs. For BERT, we used BERT-Base Uncased. Furthermore, we consider the CLS token representing the encoding of a tweet as input to a logistic regression for the classification task.

[1] https://spacy.io/.
[2] https://stanfordnlp.github.io/.
[3] https://www.nltk.org/.
[4] https://github.com/HawreH/iner_sfl.

Table 8
Performance of classifiers with different subsets of features and based on F1 measure with Macro, Weighted and Micro averaging.

	Syntagmatic				Paradigmatic			Syntagmatic + Paradigmatic
	Reference	Verb processes	Circumstances	All	Contextual cohesion	Factual cohesion	All	All
Macro average	53.42	55.87	41.9	57.88	42.98	65.06	65.69	68.62
Weighted average	61.86	62.83	54.21	64.5	54.9	69.14	69.46	72.77
Micro average	67.24	65.81	65.46	67.34	65.46	69.45	69.55	73.63

Table 9
Performance of our best classifiers built using domain-specific and combined data instances for training and testing.

Domain	Precision	Recall	F1
Person	73.72	74.86	73.52
Organization	76.72	77.19	76.08
Location	78.9	79.02	78.84
Event	72.21	71.71	71.09
Product/Device	77.39	75.7	75.32
WrittenWork	68.34	68.83	67.8
Film	70.42	69.66	69.17
All domains	72.71	73.63	72.77

5.4. Performance evaluation

We report the performance of the model trained on our proposed features and compare it against the baselines. We present performance measures both overall as well as per domain. The results can be seen in Table 7. Our results indicate that our model outperforms the baseline methods on different metrics. This is a strong indication that the proposed features defined for the task of recognizing implicit entities in tweets are suitable and effective. Our approach has a balanced performance on both weighted average precision and recall metrics, which is ≈73%. We consider this to be a strong performance considering the fact that fastText, StarSpace and the BERT-based classifier perform not much better than the performance of a random binary classifier on all metrics and even lower than a random classifier on the recall metric by the fastText classifier.

Now, in order to measure the impact of our different types of proposed features on the overall performance, we perform an ablation study with each such subset of features and report the performance in Table 8. Our experiments show that features in the paradigmatic class are the strongest set of features that are able to recognize entity implicatures more effectively. This indicates the importance of factual and contextual knowledge graph information for identifying entity implicatures in tweets. Within the paradigmatic features, those that are based on factual cohesion show a stronger performance compared to contextual cohesion features. This indicates that relying on the subset of knowledge graph relations that are temporally aligned with the content shared on Twitter with the tweet under consideration would not always be sufficient for identifying implicature in tweets. Our manual inspection of the reason for this indicates that contextual cohesion features are more effective for cases when the theme of the tweet is aligned with the general topics discussed in the social network. In such cases, the set of relations that are extracted from the pooled tweets align well with the tweet under consideration and hence lead to better performance. However, when the topic of the tweet does not align with the mainstream topics, the relations that are extracted from the pooled tweets do not align with the considered tweet and hence lead to incorrect classification.

We further note that while syntagmatic features show a weaker overall performance compared to paradigmatic features, they do have reinforcing and synergistic impact on paradigmatic features. This can be observed for instance, when comparing the weighted average performance of all features for syntagmatic, paradigmatic and syntagmatic+paradigmatic cases. As observed in Table 8, paradigmatic features show an overall performance of 69.46% while syntagmatic features report 64.5%. However, when the set of features from these two classes are put together, they show an increased performance of 72.77%, which is a notable improvement of 4.76% over the paradigmatic features. Therefore, while syntagmatic features are not as effective, they do show complimentary performance and are useful to be included in this task.

To show that the performance of our proposed features (last column in Table 8) are consistent across different domains, we report performance based on the different domains that are present in the gold standard dataset in Table 9 and only for the positive class in Table 10. As seen in the tables, the performance of the proposed features is quite consistent across different domains in terms of precision, recall and f1-score. Therefore, we conclude that our proposed approach for identifying implicit entity mention scales well to different domains regardless of the terminology or grammatical structure used in those domains.

5.5. Feature importance

We further evaluate the importance of the different feature categories for identifying implicature in tweets. To do so, we rank-order the proposed features through their Gini score. Fig. 3 depicts the number of features from either the paradigmatic or the syntagmatic categories that have appeared in the top-k list of features. As seen in the figure, four of the five features in the set of top-5 features with the highest Gini score are from the Paradigmatic feature category. This is consistent with the performance of the

Table 10
Performance of our best classifiers built using domain-specific and combined data instances for training and testing reported on **positive class only**.

Domain	Precision	Recall	F1
Person	64.15	50.49	56.19
Organization	66.78	49.58	56.09
Location	70.51	63.19	66.41
Event	59.81	47.75	51.77
Product/Device	78.4	68.58	71.68
WrittenWork	59.62	50.51	53.98
Film	64.88	54.59	59.08
All domains	62.18	50.92	55.96

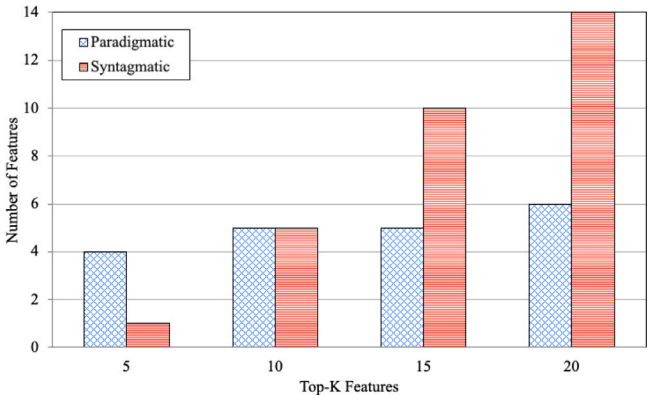

Fig. 3. The distribution of the number of features in each category of the top-k ranked features list.

ablation studies that were reported earlier where Paradigmatic features showed better overall performance. We also note that there are an equal number of features from the Paradigmatic feature category as there from the Syntagmatic category. This speaks to the fact that while Syntagmatic features are not as effective as the top-performing Paradigmatic features, they are still strong features and hence explains the synergistic impact of such features on Paradigmatic features. In the top-20 features, six features are from the Paradigmatic feature category, while fourteen are Syntagmatic features. Therefore, while there is a larger number of effective Syntagmatic features, the performance of the six Paradigmatic features are stronger.

In order to understand which feature sub-categories within the Paradigmatic and Syntagmatic categories have the most impact on performance, we additionally depicted the number of features from each of these subcategories in Fig. 4. We make several important observations from this figure. First, we find that from the features in the Paradigmatic category, factual cohesion features are the only features that appear in the set of top-5 features (in fact in the top-20 features as well). This indicates that factual cohesion features are more effective compared to contextual cohesion features. We also had a similar observation in Table 8 where factual cohesion features showed a superior performance compared to contextual cohesion features. Second, we observe that similar to Paradigmatic contextual cohesion features, Syntagmatic circumstance features do not appear in the set of top-20 features with the highest Gini score. Again, this is consistent with the lower performance of circumstance features reported in Table 8. Finally, we report that Syntagmatic verb process features are the ones that predominantly represent the Syntagmatic category in the set of top-20 features (except for two features from the corefential properties sub-category).

Overall, we conclude that:

- While there are larger number of Syntagmatic features in the top-20 features (16 features), the performance shown by the smaller set of Paradigmatic features is higher and hence leads to a stronger performance on detecting implicature in tweets;
- Within the set of Paradigmatic features, only factual cohesion features appear in the list of top-20 features, which justifies the better performance of factual cohesion features compared to contextual cohesion features;
- In the Syntagmatic features, circumstance features do not appear in the top-20 features and hence show a low performance on the task; however, features from the verb processes sub-category have the highest representation on the set of top-20 features and hence show the highest performance on the task from among the Syntagmatic features.

6. Discussion

In this section, we are interested in presenting a more in depth understanding of the strengths and weaknesses of the proposed features for the recognition of implicit entities through carefully reviewing the instances that have been labeled incorrectly by

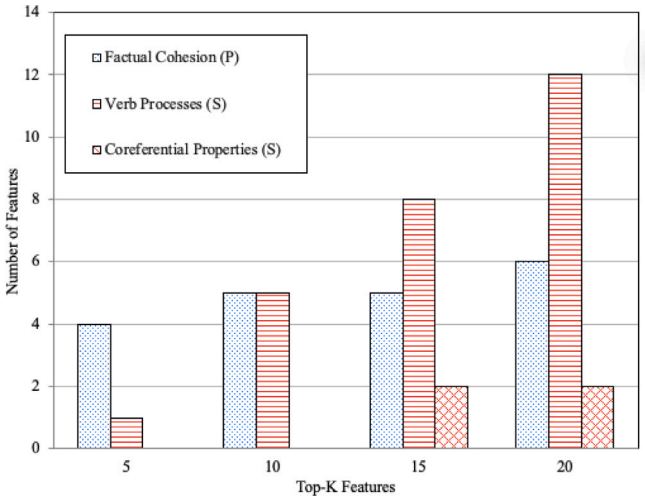

Fig. 4. The distribution of the number of features in each feature sub-category of the top-k ranked features list.

our proposed approach. For such tweets, we indicate why our proposed features fail to provide the correct classification. We have carefully reviewed every data instance where our approach has not been able to correctly provide a classification and categorized such cases in different error types. In what follows, we elaborate on each type of error:

Error Type 1: *Issues with Coreference Properties*: As indicated by our results, the set of features in the coreferential properties subset are effective in recognizing implicit entities. However, several types of errors may happen when it relates to this feature sub-category:

- First, a lack of pronouns will lead to situations where our work is not able to identify the presence of coreference. For instance, one of the most common forms of implicit entity mentions appears in the following tweet: "Deadly Car Bomb Blast Inside Afghanistan's Capital City - Primary World Afghanistan New". In this tweet, given that the text does not include any pronouns, our system fails to recognize the implicit mention.
- Second, our work relies heavily on identifying pronoun referents. In cases when we are not able to link a pronoun to a referent, our work will not be able to function properly. As an instance, in the tweet "Jordan Belfort is quickly becoming my idol after reading his book Real Men LearnFromTheirMistakes", we were not able to link the coreferent of the pronoun "his" properly and therefore, our work was not able to properly identify implicature in this tweet.
- Finally, the other type of coreference-related error happens in our work for cases when the pronoun does not necessarily point to a referent. This typically happens for pronouns that are general in nature, such as the pronoun "it" in tweets such as "It's a beautiful day".

Error Type 2: *Misleading Explicit Entities*: Several of our proposed features rely on the explicit entities found within the tweet. In such features, we depend on the explicit entity to determine implicature. There are different errors that can happen when dealing with explicit entities:

- It is common that the presence of an explicit entity is missed within a tweet. As an example in the tweet "Workforce, skills and R&D capacity - 3 reasons Ireland is a business model built to last, according to Edel Creely", although quite obvious to the human observer, explicit entity linkers miss 'Ireland' as an explicit entity. As a result of missing the explicitly mentioned entity, our work determines that this tweet consists of an implicit mention while the entity was explicitly indicated.
- Another issue happens when an incorrectly linked entity impacts the quality of the values extracted for our features. As an example, in the tweet "OMG !!! I think I'm gonna cry with the Stephen Hawking movie. Eddie Redmayne will be Stephen Hawking and David Thewlis is here as well !!" In this tweet, the word "will" has incorrectly been identified as "Will Smith", creating non aligned signals to the implicit mention of the tweet, i.e., "The Theory of Everything" movie. This source of error is difficult to resolve as explicit entity linking in tweets is challenging and existence of wrongly linked entities is inevitable.
- A final entity related error is typing. A number of our proposed features leverage typing information of explicit entities found within the tweet. For instance, feature FC2 is extracted based on a mapping between entity types and category-related keywords. Two major types of error can happen with this regard: (1) when an explicit entity's type cannot be resolved correctly from a KB, and (2) when an explicit entity's fine-grained type cannot be related to its coarse-grained type. The first issue is not very prevalent, and often adopting multiple KBs can help address the issue. As to the second issue, a comprehensive list of fine- and coarse-grained mapping can help resolve this issue, where for each fine-grained type (e.g., musician) there would be a coarse-grained type based on a dataset taxonomy (e.g., Person). The development of such a taxonomy would be a difficult manual endeavor and hence taxonomy learning techniques (Al-Aswadi, Chan, & Gan, 2020; Shang, Zhang, Liu, Li, & Han, 2020) could be helpful in this space.

Error Type 3:*Inaccurate Relation Extraction*: The correct extraction of relation types and instances is important in our work as it impacts the value of some of our proposed features. We find that there are several issues as it pertains to relation extraction.

- First, open information extraction does not always show reasonable effectiveness when identifying relations from tweets due to several reasons, including lack of proper punctuation and complexity of the sentences, and lack of grammatical structure, among others. For instance, in one of the tweets, i.e., '*I suspect this won't be a GREAT movie, but it was a pretty wild book — by Joe Hill, son of Stephen King.*', our work has not been able to extract the relation between *Joe Hill* and *book*. This results in inaccurate extraction of several features. As an instance, Factual Cohesion features investigate relation triples inside text that align with a KB relation, and whether any explicit entities are missing. Here, the KB triple (*Joe Hill, dbo:author, dbr:Horns_(novel)*) can be a strong signal that there is an implicit entity inside the tweet upon successful extraction of the relation between Joe Hill and wild book, which in this case has not been identified.
- Another related challenge is when the open information extraction method manages to identify relationship triples but the alignment of the identified relationship with KB triples turns out to be non-trivial. For instance, in the previous example, the output from the open information extraction module would be (*Joe Hill*, *by*, *wild book*). Such a triple is represented as (*Joe Hill*, *dbo:author*, *wild book*). Aligning "*by*" with "*dbo:author*" is a difficult task and in itself can potentially be an independent research problem.

7. Concluding remarks

In this paper, we have adopted a systematic feature engineering approach for performing the task of implicit entity recognition. We have leveraged Systemic Functional Linguistics as our theoretical framework in order to model the tweet environment for the purpose of identification of implicit named entities when they happen. In doing so, we have systematically designed two broad categories of features, namely paradigmatic and syntagmatic. The designed features are inspired by SFL's view point on meaning formation in language, where there is an emphasis on the social and conceptual roles of language. The syntagmatic features, leveraging syntagmatic linguistic tools as introduced by SFL, capture the local signals of implication. On the other hand, the paradigmatic class of features are meant to capture patterns of implication through global characteristics, such as relationships between explicit entities within text and other textual components. Our experiments indicate that our system based on the proposed features is able to outperform the baseline methods for implicit entity recognition.

Supported by our empirical results, we argue that implicit named entity recognition requires deeper levels of natural language understanding than its explicit counterpart. As such, the choice of the Systemic Functional Linguistics framework is appropriate as it allows us to capture how meaning is formed within natural language and hence can support the identification of cases when implicature happens. Within our features, we find that paradigmatic features are more effective than syntagmatic features pointing to the importance of knowledge graph information for identifying entity implicatures in tweets. We further find that within paradigmatic features, those that depend on factual cohesion are far more effective than contextualized features. This is supported by evidence from both overall model performance as well as feature importances based on the Gini score. Within the context of Syntagmatic features, we find that circumstance features show the weakest performance and have the lowest contribution to the performance of the implicature detection task. On the other hand, features from the verb processes sub-category show the largest number of presence in the set of most important top-20 features. Finally, while we find that paradigmatic features are the most effective features for detecting implicature, and that syntagmatic features have lower performance, the two categories of features have synergistic impact and lead to increased performance when used in tandem.

We are excited about several future directions of research based on the work in this paper. One major line of research will involve designing algorithmic approaches that while leveraging insights from the designed features and their importances, will endeavor to relieve the current shortcomings as elaborated in the error types in the previous section. These can fall in the following research areas:

1. End to end models: As can be seen from the error analysis, our system relies on modules that are detached and work in isolation. Such critical modules include relation extraction, explicit entity linking and typing, and coreference resolution. There have been works in the literature that emphasize effectiveness of tackling these problems in tandem as they provide useful signals. Following this intuition, an interesting approach to tackle the recognition of implicit entities would be to investigate the possibility of recognizing implicit mentions of entities through a novel problem definition for the aforementioned problems. A major advantage of such an approach, in addition to simplicity and maintainability, would potentially be less information loss and less noisy signals. As a concrete example, information is often lost when aligning the open information extraction module's output with KB triples. Such alignment can be learned in the model jointly without losing critical information;
2. We are interested in exploring solutions based on zero-shot and few-shot approaches. As mentioned earlier, a major bottleneck for the task of implicit entity recognition and linking is lack of labeled data. Unfortunately, this lack of data for implicit phenomena and underrepresented language is prevalent. Apart from unsuitability of many of the existing approaches to solving similar problems, this lack of data limits the generalizability in this space. As labeling huge amounts of data will be expensive, few-shot algorithms, or generally speaking unsupervised and semi-supervised approaches that can potentially leverage more data for training, may have better generalizability; and,

3. Finally, we are interested in benefiting from the broadly indicated power of pre-trained language models. Such LMs have already been leveraged in explicit entity recognition and linking (Li, Zhang, & Zhou, 2020; Liang et al., 2020). However, a major drawback for extending their capability to implicit entity recognition and linking is the fact that implicit mentions do not have surface forms and lack textual placeholders. Additionally, fine-tuning such models and/or validating their performance requires a considerable amount of labeled data, which are currently not available in this field.

CRediT authorship contribution statement

Hawre Hosseini: Conceptualization, Methodology, Software, Experiments and analysis, Writing – original draft. **Mehran Mansouri:** Conceptualization, Methodology. **Ebrahim Bagheri:** Supervision, Conceptualization, Methodology, Writing – review & editing.

References

Al-Aswadi, F. N., Chan, H. Y., & Gan, K. H. (2020). Automatic ontology construction from text: a review from shallow to deep learning trend. *Artificial Intelligence Review*, *53*(6), 3901–3928.

Bateman, J. A., Kasper, R. T., Moore, J. D., & Whitney, R. A. (1990). *A general organization of knowledge for natural language processing: the penman upper model*: *Technical report*, USC/Information Sciences Institute, Marina del Rey, CA.

Berger, A., Della Pietra, S. A., & Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, *22*(1), 39–71.

Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *Fifth conference on applied natural language processing* (pp. 194–201). Washington, DC, USA: Association for Computational Linguistics, http://dx.doi.org/10.3115/974557.974586, URL https://aclanthology.org/A97-1029.

Botzer, N., Ding, Y., & Weninger, T. (2021). Reddit entity linking dataset. *Information Processing & Management*, *58*(3), Article 102479.

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, *4*, 357–370.

Das, S., & Paik, J. H. (2021). Context-sensitive gender inference of named entities in text. *Information Processing & Management*, *58*(1), Article 102423.

Elke, T. (Cassel, London, 1999). *Systemic functional grammar in natural language generation*. London: Continuum International Publishing Group.

Fawcett, R. P., & Tucker, G. H. (1990). Demonstration of GENESYS: A very large, semantically based systemic functional generator. In *COLING 1990 Volume 1: Papers presented to the 13th international conference on computational linguistics*.

Ferragina, P., & Scaiella, U. (2010). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 1625–1628). ACM.

Ganea, O.-E., & Hofmann, T. (2017). Deep joint entity disambiguation with local neural attention. arXiv preprint arXiv:1704.04920.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42.

Geurts, P., & Louppe, G. (2011). Learning to rank with extremely randomized trees. In *Proceedings of the learning to rank challenge* (pp. 49–61). PMLR.

Gupta, N., Singh, S., & Roth, D. (2017). Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2681–2690).

Halliday, M. A. K., & Matthiessen, C. M. (2013). *Halliday's introduction to functional grammar*. Routledge.

Hosseini, H. (2019). Implicit entity recognition, classification and linking in tweets. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 1448–1448).

Hosseini, H., & Bagheri, E. (2020). From explicit to implicit entity linking: A learn to rank framework. In *Canadian conference on artificial intelligence* (pp. 283–289). Springer.

Hosseini, H., & Bagheri, E. (2021). Learning to rank implicit entities on Twitter. *Information Processing & Management*, *58*(3), Article 102503.

Hosseini, H., Nguyen, T. T., & Bagheri, E. (2018). Implicit entity linking through ad-hoc retrieval. In *2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)* (pp. 326–329). IEEE.

Hosseini, H., Nguyen, T. T., Wu, J., & Bagheri, E. (2019). Implicit entity linking in tweets: An ad-hoc retrieval approach. *Applied Ontology*, *14*(4), 451–477.

Huang, L., Yuan, B., Zhang, R., & Lu, Q. (2020). Towards linking camouflaged descriptions to implicit products in E-commerce. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 901–910).

Ibrahim, Y., Amir Yosef, M., & Weikum, G. (2014). Aida-social: Entity linking on the social stream. In *Proceedings of the 7th international workshop on exploiting semantic annotations in information retrieval* (pp. 17–19).

Jie, Z., & Lu, W. (2019). Dependency-guided LSTM-CRF for named entity recognition. arXiv preprint arXiv:1909.10148.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: volume 2, short papers* (pp. 427–431). Valencia, Spain: Association for Computational Linguistics, URL https://aclanthology.org/E17-2068.

Kasper, R. T. (1988). An experimental parser for systemic grammars. In *Coling budapest 1988 volume 1: international conference on computational linguistics*.

Kay, M. (1985). Parsing in functional unification grammar. *Natural Language Parsing*, 251–278.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 260–270). San Diego, California: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N16-1030, URL https://aclanthology.org/N16-1030.

Li, X., Zhang, H., & Zhou, X.-H. (2020). Chinese clinical named entity recognition with variant neural structures based on BERT methods. *Journal of Biomedical Informatics*, *107*, Article 103422.

Liang, C., Yu, Y., Jiang, H., Er, S., Wang, R., Zhao, T., et al. (2020). Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1054–1064).

Logeswaran, L., Chang, M.-W., Lee, K., Toutanova, K., Devlin, J., & Lee, H. (2019). Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 3449–3460). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-1335, URL https://aclanthology.org/P19-1335.

McCord, M. C. (1975). On the form of a systemic grammar. *Journal of Linguistics*, *11*(2), 195–212.

McCord, M. C. (1977). Procedural systemic grammars. *International Journal of Man-Machine Studies*, *9*(3), 255–286.

Meij, E., Weerkamp, W., & De Rijke, M. (2012). Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on web search and data mining* (pp. 563–572). ACM.

Mendoza, M., Parra, D., & Soto, A. (2020). GENE: Graph generation conditioned on named entities for polarity and controversy detection in social media. *Information Processing & Management*, *57*(6), Article 102366.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning–based text classification: a comprehensive review. *ACM Computing Surveys*, *54*(3), 1–40.

Mohit, B. (2014). Named entity recognition. In *Natural language processing of semitic languages* (pp. 221–245). Springer.

Murty, S., Verga, P., Vilnis, L., Radovanovic, I., & McCallum, A. (2018). Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers)* (pp. 97–109).

Nozza, D., Manchanda, P., Fersini, E., Palmonari, M., & Messina, E. (2021). LearningToAdapt With word embeddings: Domain adaptation of named entity recognition systems. *Information Processing & Management*, *58*(3), Article 102537.

O'Donnell, M. (1993). Reducing complexity in a systemic parser. In *Proceedings of the third international workshop on parsing technologies* (pp. 203–218).

O'Donnell, M., & Bateman, J. A. (2005). Sfl in computational contexts: a contemporary history. *Continuing Discourse on Language: A Functional Perspective*, *1*, 343–382.

Perera, S., Mendes, P. N., Alex, A., Sheth, A. P., & Thirunarayan, K. (2016). Implicit entity linking in tweets. In *European semantic web conference* (pp. 118–132).

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 1524–1534).

Shang, J., Zhang, X., Liu, L., Li, S., & Han, J. (2020). Nettaxo: Automated topic taxonomy construction from text-rich network. In *Proceedings of the web conference 2020* (pp. 1908–1919).

Souza, F., Nogueira, R., & Lotufo, R. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. In *Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20-23, 2020, Proceedings, Part I* (pp. 403–417). Berlin, Heidelberg: Springer-Verlag, ISBN: 978-3-030-61376-1, http://dx.doi.org/10.1007/978-3-030-61377-8_28.

Steiner, E. (1987). *The development of the EUROTRA-D system of semantic relations*. IAI.

Thompson, G. (2013). *Introducing functional grammar*. Routledge.

Wilcock, G. (1993). *Interactive Japanese-European text generation-an approach to multilingual export translation based on Systemic Functional Grammar*. Citeseer.

Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., & Weston, J. (2018). Starspace: Embed all the things!. In *Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1*.

Wu, L., Petroni, F., Josifoski, M., Riedel, S., & Zettlemoyer, L. (2019). Scalable zero-shot entity linking with dense entity retrieval. In *22nd conference on computational natural language learning*.

Zeng, D., Sun, C., Lin, L., & Liu, B. (2017). LSTM-CRF For drug-named entity recognition. *Entropy*, *19*(6), 283.

Zhao, X., Wang, D., Zhao, Z., Liu, W., Lu, C., & Zhuang, F. (2021). A neural topic model with word vectors and entity vectors for short texts. *Information Processing & Management*, *58*(2), Article 102455.