Troubleshooting and Optimizing Named Entity Resolution Systems in the Industry

Panos Alexopoulos(✉), Ronald Denaux, and Jose Manuel Gomez-Perez

Expert System Iberia, Madrid, Spain
{palexopoulos,rdenaux,jmgomez}@expertsystem.com

Abstract. Named Entity Resolution (NER) is an information extraction task that involves detecting mentions of named entities within texts and mapping them to their corresponding entities in a given knowledge resource. Systems and frameworks for performing NER have been developed both by the academia and the industry with different features and capabilities. Nevertheless, what all approaches have in common is that their satisfactory performance in a given scenario does not constitute a trustworthy predictor of their performance in a different one, the reason being the scenario's different characteristics (target entities, input texts, domain knowledge etc.). With that in mind, in this paper we describe a metric-based Diagnostic Framework that can be used to identify the causes behind the low performance of NER systems in industrial settings and take appropriate actions to increase it.

1 Introduction

Information Extraction (IE) involves the automatic extraction of structured information from texts, such as entities and their relations, in an effort to make the information of these texts more amenable to applications related to Question Answering, Information Access and the Semantic Web. In turn, Named Entity Resolution (NER) is an IE subtask that involves detecting mentions of named entities within texts (e.g. people, organizations or locations) and mapping them to their corresponding entities in a given knowledge source. The typical problem in this task is ambiguity, i.e. the situation that arises when a term may refer to multiple different entities. For example, the term "Tripoli" may refer, among others, to the capital of Libya or to the city of Tripoli in Greece. Deciding which reference is the correct one is the primary challenge for NER systems.

In the last years, systems and frameworks for performing NER have been developed both by the academia and the industry with different features and capabilities [1,5–8,10,15]. These systems typically vary in a number of dimensions, including the type of background domain knowledge they utilize (annotated corpora, thesauri, ontologies etc.), the algorithms they apply, and their customization capabilities, i.e., the ability provided to the user to change key parameters of the system so as to adapt it to his/her particular domain and/or application scenario. Moreover, the effectiveness of several NER systems has been

F. Gandon et al. (Eds.): ESWC 2015, LNCS 9088, pp. 559–574, 2015.
DOI: 10.1007/978-3-319-18818-8_34

empirically measured and reported in their respective scientific publications as well as in dedicated evaluation papers [4,14].

In our opinion, the most interesting aspect of these evaluations is not so much the absolute precision and recall scores that each system achieves but rather the volatility of these scores as the characteristics of the problem (texts to be analyzed, available domain knowledge etc.) change. For example, in [6] the effectiveness of the AIDA NER system is found to be 83 % on the AIDA-YAGO2 dataset and 62 % on Reuters-21578. Similarly, in [10], the effectiveness of DBPedia Spotlight is found to be 81 % when applied on a set of 155,000 wikilink samples and 56 % on a set of 35 paragraphs from New York Times documents. In another paper [15] Spotlight achieves an F1 score of 34 % on the AIDA/CO-NLL-TestB dataset (created in [6]). Finally, the AGDISTIS system [15] scores 76 % on the AQUAINT dataset (created in [12]), 60 % on the AIDA/CO-NLL-TestB dataset and 31 % on the IITB dataset (created in [9]).

What these scores illustrate is that **a NER system's satisfactory performance in a given scenario does not constitute a trustworthy predictor of its performance in a different one**. Or, to put it differently, it's always likely that the system will perform poorly when the scenario's characteristics change. This is an important ramification for developers of NER solutions in the industry as commercial clients typically expect a high and consistent performance from the systems they pay for. Thus, a question that naturally arises is the following: **If in a given NER scenario the system's effectiveness is found to be low, what can be done in order to increase it?**

In an effort to answer this question we describe in this paper a **NER Diagnostic Framework** that consists of a set of metrics that quantify particular aspects of both the problem and the solution applied in a given scenario (such as for example the quality of the system's knowledge base). The idea is that via the calculation and interpretation of these metrics, NER developers are able to identify the most likely causes of their system's low performance and act on this information in order to increase it. In this paper we describe in detail the framework's metrics and we provide illustrative examples of their application and usefulness in a number of concrete cases.

The structure of the rest of the paper is as follows. In the next section we provide a high-level view of the way NER systems work and we use this view in order to identify the potential reasons why such systems might not be effective. In Sect. 3 we define a set of metrics that can be used to troubleshoot a NER system, i.e., to determine (i) which of these reasons and to what extent apply in a given scenario and (ii) the necessary actions for dealing with these reasons and reducing their effect. Section 4, in turn, describes how the application of the diagnostic framework enabled us to achieve significantly increased NER effectiveness in two different cases. Finally, in Sects. 5 and 6 we make a critical discussion of our work, summarize its key aspects and outline the potential directions it could take in the future.

2 NER Systems and Their Effectiveness

2.1 Anatomy of a NER System

As suggested in the introduction, a NER system detects mentions of entities in texts and maps them unambiguously to their corresponding entities in a given knowledge resource. To do that, the system typically utilizes four types of input (Fig. 1):

1. A set of texts on which NER is to be performed.
2. A set of target entities which are to be detected and disambiguated.
3. An entity thesaurus where each entity is associated to a unique identifier and a set of potential surface forms.
4. Some knowledge resource to serve as contextual evidence for the disambiguation of ambiguous entity mentions in the texts.

The latter input is derived from the strong contextual hypothesis of Miller and Charles [11] according to which terms with similar meanings are often used in similar contexts. For a given entity, such a context usually consists of (i) the words that "surround" the entity in some reference text [3,10] and/or (ii) the entities that are related to this entity in some knowledge graph [6,8,15]. Thus, for example, a disambiguation context for the entity "Larry Page" could be entities like "Google" and "PageRank" whereas for the entity "Jimmy Page" entities like "Led Zeppelin" and "Hard Rock". Consequently, knowledge resources that contain such contexts (and thus used by NER systems) are texts that are already annotated with these entities (e.g., wikipedia articles) as well as entity-related knowledge graphs (e.g., DBPedia[1] or YAGO[2]). Given these inputs, a NER system works in two steps:

- **Step 1:** The entity thesaurus and some NLP framework (e.g., GATE[3]) are used to extract from the texts terms that possibly refer to entities. The result is a set of terms, each associated to a set of candidate entities.
- **Step 2:** The contextual evidence knowledge resource is used to determine for each term the most probable entity it refers to (disambiguation).

In the second step, when the evidence knowledge resource consists of annotated texts, disambiguation is performed by calculating the similarity between the term's textual context in the input text and the contexts of its candidate entities in the annotated texts. When the contextual evidence is a knowledge graph then graph-related measures are employed to determine the similarity between the graph formed by the entities found within the ambiguous term's textual context and the sub-graphs formed by each candidate entity's "neighbor" entities. In all cases, the candidate entity with the most similar context is assumed to be the correct one.

[1] http://dbpedia.org.
[2] www.mpi-inf.mpg.de/yago-naga/yago/.
[3] http://gate.ac.uk/.

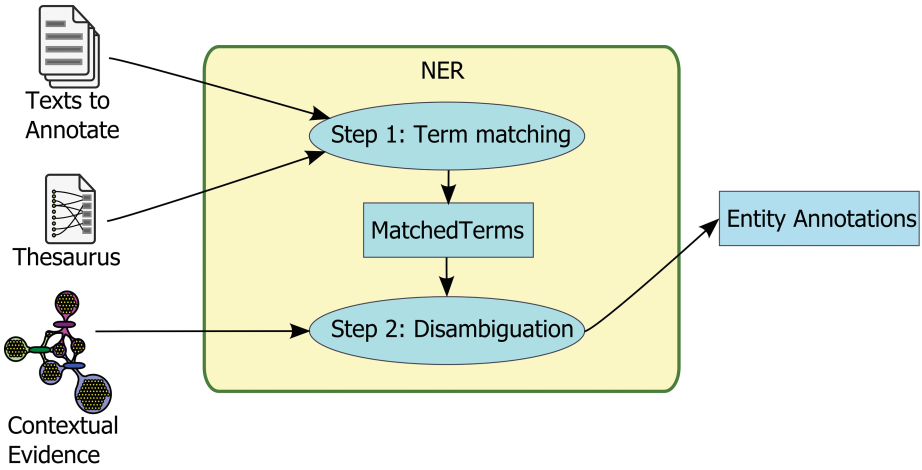

Fig. 1. Anatomy of a NER system, including inputs, processing steps and output.

2.2 When NER Effectiveness is Low and Why

Effectiveness of NER systems is typically measured in terms of *precision* and *recall*. Precision is determined by the fraction of correctly resolved terms (i.e., terms for which the entity with the highest confidence is the correct one) to the total number of detected terms (i.e., terms with at least one associated entity). Recall, on the other hand, is determined by the fraction of correctly resolved terms to the total number of existing entity mentions in the input texts. Thus, a NER system has **low precision** if the input texts do not really contain (most of) the system-assigned entities. This usually happens when:

- There is a **high degree of ambiguity**, i.e., many entities from the thesaurus are (wrongly) associated to many text terms.
- The **contextual knowledge is inadequate** to correctly fulfil the disambiguation of terms. For example, if a text contains the term "Page" as a reference to the entity "Jimmy Page" and the contextual evidence knowledge resource has no information about this entity, then the disambiguation will most likely fail.

On the other hand, a NER system has **low recall** when it fails to detect entities in the texts that are actually there. This may happen in two cases:

- When the **thesaurus is incomplete** by not containing either several of the target entities or adequate surface forms for them (e.g., the thesaurus may not associate the surface form "Red Devils" to the football team of Manchester United).
- When the system, in order to be confident about a term's disambiguated meaning, requires a certain minimum amount of contextual evidence to be found in the input texts but fails to do so. This failure may be due to the **lack of evidence** in the evidence knowledge resource and/or the texts themselves.

Obviously, the above analysis is still too abstract to be of much use; therefore our approach for troubleshooting a NER system when precision and/or recall are found to be low involves using concrete metrics to make terms like "high ambiguity" and "adequate evidence" more precise and determine the extent to which they apply in the scenario at hand.

3 Metric-based Diagnosis of NER Systems

3.1 Ambiguity Metrics

In the previous section, we mentioned that one reason for low NER precision is high ambiguity, i.e., the wrong association of many target entities to terms in the text. This may happen when many target entities can be mixed (i.e., confused) with:

- **Common lexical terms** that are not really entities. For example, if the target entities are companies then the location services company "Factual" can be easily mixed in a text with the namesake common adjective. For the purposes of this paper we call this phenomenon **Lexical Ambiguity**.
- **Other target entities**. For example, if the target entities are locations then the city of Tripoli in Greece may be mixed with Tripoli in Libya. For the purposes of this paper we call this phenomenon **Target Entity Ambiguity**.
- **Non-target entities** from the contextual evidence knowledge graph. For example, if the target entities are football teams and the knowledge graph contains also locations then the *team of Barcelona* may be mixed with the *city of Barcelona*. For the purposes of this paper we call this phenomenon **Knowledge Graph Ambiguity**.
- **Entities from other domains**, not included in the thesaurus nor in the evidence knowledge graph. For example, if the target entities are companies then the telecom company Orange may be mixed with the namesake fruit. For the purposes of this paper we call this phenomenon **Global Ambiguity**.

In order to identify which of these four ambiguity types and to what extent characterize a given NER scenario, we work as follows. First, we consider a representative sample of the texts that we are supposed to perform NER on and we manually annotate them with target entities as well as non-target entities from the contextual evidence knowledge graph. Subsequently, we perform the same task in an automatic way by using the NER system *without any disambiguation* (i.e., we perform only step 1 of the process described in Sect. 2.1). Having done that we may measure the different types of ambiguity as follows:

- **Lexical Ambiguity:** We measure this as the percentage of terms which (i) are common lexical terms rather than entities in the text, (ii) have not been manually annotated with any target entity and (iii) have been wrongly mapped by the system to one or more target entities.

- **Target Entity Ambiguity:** We measure this as the percentage of terms which (i) have been annotated with a target entity and (ii) have been mapped by the system to this target entity but also to other target entities.
- **Knowledge Graph Ambiguity:** We measure this in two complementary ways. First, as the percentage of terms which (i) have been manually annotated with a target entity and (ii) have been mapped by the system to this target entity but also to other non-target entities. For the purposes of this paper we call this KGA_1. Second, as the percentage of terms which (i) have been manually annotated with a non-target entity and (ii) have been mapped by the system to this entity but also to other target entities. For the purposes of this paper we call this KGA_2.
- **Global Ambiguity:** We measure this as the percentage of terms which (i) are not common lexical terms but actual entities in the texts, (ii) have not been manually annotated with any entity and (iii) have been mapped by the system to one or more target entities.

All the above percentages are calculated over the total number of terms the NER system has detected in the texts. Also, please note that the above ambiguity types and metrics are not meant to replace any existing formal ambiguity classification frameworks [2]; they are merely informal tools which, as we will show in subsequent sections, we have found to be very useful in analyzing NER scenarios.

3.2 Evidence Adequacy Metrics

Complementary to high ambiguity, a second reason for low NER effectiveness is the inadequacy of the contextual knowledge applied as disambiguation evidence (step 2 of the process described in Sect. 2.1). When this knowledge has the form of a knowledge graph, then by adequacy we practically mean two things:

1. How rich is the knowledge graph in terms of relation/attribute values for its entities. As suggested in Sect. 2.1, these values are used as contextual disambiguation evidence, therefore if many entities lack them, their disambiguation will probably fail. For example, if we want to disambiguate film mentions in texts, a potential evidence could be the actors that played in them. If this relation is poorly populated in the knowledge graph, then the latter may be inadequate for the particular task.
2. How prevalent is the contextual evidence provided by the knowledge graph in the input texts. Even if the knowledge graph is rich, it won't help if the texts do not contain the evidence it provides. Considering the film example, even if we know all the film's actors, this knowledge will not be useful if films and their actors do not co-occur in the texts.

Knowledge graph richness can be measured in many ways, depending on the desired level of detail. Some metrics we have found useful are the following:

- **The percentage of target entities with no related entities at all.** If this number is high then the knowledge graph is practically useless for the disambiguation of the particular entities.
- **The average number of entities a target entity is related to.** If this number is lower than expected (e.g., if films are related in average to only 1 or 2 entities when they are typically expected to be related to several actors, directors, producers, characters etc.) then the knowledge graph might not be as useful as it could.
- **The average number of entities a target entity is related to via a specific relation.** If this number is lower than expected then this relation cannot really contribute to the disambiguation task even if it is expected to do so. For example, if the "hasActor" relation for films is poorly populated (e.g., only one or two actors per film) then the NER system is practically not able to use any actor mentions in the texts as film disambiguation evidence.

The above metrics can be easily calculated by merely querying the knowledge graph. On the other hand, in order to measure the prevalence of the graph's contextual evidence in the input texts we use both the texts and the graph. In particular, we consider again the representative sample of input texts that we used for measuring ambiguity in the previous section and which we have already manually annotated with target entities as well as non-target entities from the contextual evidence knowledge graph. Then, for each pair of a target and non-target entity in the annotated texts, we derive from the knowledge graph the relation(s) and/or the relation paths (up to a certain length) through which the entities are linked. This allows us to calculate the following:

- **The percentage of target entities for which there is at least one evidential entity in the texts.** If this number is low then obviously the knowledge graph is not useful for the given texts.
- **The average number of evidential entities a target entity is related to in the texts.** If this number is too low then again the knowledge graph is not appropriate for the given text.
- **The percentage of target entities for which there is at least one evidential entity in the texts via a specific relation or relation path.** If this number is low then this particular relation is not useful for the given texts.
- **The average number of evidential entities a target entity is related to in the texts via a specific relation (or relation path).** Again, this number allows to assess the relative usefulness of the graph's relations for the disambiguation task.

Please note that the definition of metrics for the adequacy of text-based evidence knowledge resources (i.e., entity annotated texts) has been left out of this paper's scope and will be addressed in future work. The reason for that is that we haven't used so much this kind of evidential knowledge in the NER scenarios we have come against so far.

3.3 Acting on the Metrics

As suggested in the introduction, the ultimate goal of the metrics is to enable practitioners to improve the unsatisfactory effectiveness of a NER system in a given scenario. For that, in this section, we map the potential values of these metrics to concrete actions that may achieve this goal.

For starters, if the Lexical Ambiguity of the entities is considerable then the word sense disambiguation (WSD) capabilities of the linguistic analysis component of the NER system need to be enhanced. Depending on the existing capabilities of the system and the extent of the problem, these enhancements can range from simple heuristics (e.g., that a company mention in a text typically starts with a capital letter) to complete implementations of WSD frameworks [13].

On the other hand, if Global Ambiguity is found to be high, then it may be that many of the input texts are not really related to the domain of the target entities. For example, if NER is performed on news articles in order to detect mentions of films (with e.g., LinkedMDB[4] as an evidence knowledge graph) and most of these articles are not relevant to the cinema domain, then it's quite likely that many non-film entities will be mistaken for films. To remedy this situation one could possibly expand the evidence knowledge graph so as to include all the domains the input texts are about; nevertheless this can be quite difficult and resource-intensive to achieve. Another, more practical approach, would be to use a domain/topic classifier in order to filter out the non-relevant texts and apply the NER process only to the relevant ones. Intuitively, this will boost precision even if some level of recall is sacrificed.

The next metric that can lead to action is the Knowledge Graph Ambiguity, i.e. the ambiguity between target entities and entities from the evidence knowledge graph. As suggested in Sect. 3.1 we measure this by means of two different percentages, i.e., the percentage of text target entities that may be confused with evidential ones (KGA_1) and the percentage of text evidential entities that may be confused with target ones (KGA_2). If KGA_1 is found to be high and KGA_2 low, then what is most probably needed is the pruning of the evidence knowledge graph in order to remove parts of it that are not so essential but can still cause noise.

To show why pruning may be necessary assume that we perform NER in a set of film reviews, targeting mentions of actors and using DBPedia as an evidence knowledge graph. Since DBPedia contains many person entities that are not actors, it is quite likely that many actor mentions in the texts will be mistaken for other persons (e.g., the actor Roger Moore could be mistaken with the namesake computer scientist). A high KGA_1 score would clearly illustrate this. On the other hand, since the input texts are primarily about films, the probability that the term "Roger Moore" actually refers to the computer scientist rather than the actor is low. Again, a low KGA_2 would make this obvious. Thus, if we were to remove from the knowledge graph all the non-actor person entities, we would most likely increase precision by allowing the NER system to focus only on the disambiguation of actor entities.

[4] http://data.linkedmdb.org.

Pruning the knowledge graph may be also helpful when the latter contains misleading evidential relations. For example, consider an excerpt from a contemporary football match description saying that "Ronaldo scored two goals for Real Madrid". To disambiguate the term "Ronaldo" in this text using DBpedia, the only contextual evidence that can be used is the entity "Real Madrid". Yet, there are two players with that name that are semantically related to it, namely Cristiano Ronaldo (current player) and Ronaldo Luis Nazario de Lima (former player). Thus, if both relations are considered then the term will not be disambiguated. Yet, the fact that the text describes a contemporary football match suggests that, in general, the relation between a team and its former players is not expected to appear in it. Thus, for such texts, it would make sense to ignore this relation in order to achieve more accurate disambiguation.

The pruning of the knowledge graph in the above cases can be done in two stages. In the first stage, the entities (and their relations) that are not related (directly or indirectly) to the target entities could be discarded. In the second stage, the removed entities would include those that are related to the target entities but via relations that are not prevalent in the texts. For the latter, the third knowledge graph prevalence metric of Sect. 3.2 could be used, namely the average number of evidential entities a target entity is related to in the texts via a specific relation. The pruning should start from the relations with the lowest score.

Of course, this whole exercise is meaningful only if the evidence knowledge graph has some highly prevalent relations to retain after the pruning. If that's not the case, then the ideal action would be to change/expand the knowledge graph with different relations than the ones it already has and which are most likely to appear in the texts. If that's not possible, an alternative action that could be performed in case of low graph prevalence would be the reduction of the minimum evidence threshold that the system uses in the disambiguation phase, provided however that Target Entity Ambiguity and Knowledge Graph Ambiguity are also low. This action would potentially increase recall (since much less non-ambiguous entities for which little evidence has been found in the text would be rejected by the system) without decreasing much precision (since for the few entities that are ambiguous there was not much evidence to use in the first place).

Finally, if the richness of the knowledge graph is low, the obvious thing to do would be to enrich it. Since that may not be always possible due to lack of resources, the relation prevalence metric could also be used here in order to select to enrich only the most useful relations.

Table 1 summarizes the key points of the above analysis by providing a map between observed metric values, problem diagnosis and recommended action(s). In all cases, it should be made clear that the whole framework we are describing here is characterized by some degree of inexactness, meaning that there's always a possibility that (i) a diagnosis is wrong even if the metrics support it and (ii) that the execution of a recommended action fails to improve NER effectiveness even if the diagnosis is relatively accurate. For that, every time an action is

taken, precision and recall of the NER process needs to be re-measured in order to verify that the system actually performs better. The re-measurement should be done every time with a new test set so as to ensure that our actions have not introduced any bias to the process.

4 Framework Application Cases

In this section we describe two cases where the application of the paper's diagnostic framework helped us to significantly increase the (initially low) effectiveness of Knowledge Tagger, our in-house developed NER system. Knowledge Tagger uses primarily ontological knowledge graphs as disambiguation evidence.

4.1 Case 1: Resolving Players in Football Texts

In this case we had to semantically annotate a set of textual descriptions of football match highlights from the Spanish Liga, like the following: *"It's the 70th minute of the game and after a magnificent pass by Pedro, Messi managed to beat Claudio Bravo. Barcelona now leads 1-0 against Real."*. The descriptions were used as metadata of videos showing these highlights and our goal was to determine, in an unambiguous way, which were the players mentioned in each video. The annotated descriptions were then to be used as part of a semantic search application where users could retrieve videos that showed their favorite player, with much higher accuracy.

Our first attempt towards performing this task involved using Knowledge Tagger with DBPedia as both an entity thesaurus (as it included all football players we were interested in) and an evidential knowledge graph. The result of this was a precision of 60 % and a recall of 55 %, measured against a manually annotated set of 100 texts. For comparison purposes, we also applied the AIDA NER system (that uses the YAGO knowledge graph) on the same texts and we got similar figures (precision 62 % and recall 58 %).

To diagnose the reasons for this rather mediocre performance, we calculated the metrics of Sect. 3 using a 100 text diagnostics dataset. As shown in Table 2, the main types of ambiguity that characterized our case were Target Entity Ambiguity (several players with similar names) and Knowledge Graph Ambiguity (several players sharing similar names with other DBPedia entities). In particular, KGA_1 (actual players mixed with non-players) was high while KGA_2 (actual non-players mixed with players) was low. This was rather expected as the input texts were very domain specific and thus unlikely to contain many person entities that are not footballers.

Given these metric values, we went on to prune the knowledge graph (as suggested in Sect. 3.3) by removing most of the non-football related entities as well as several player relations that had no evidential value. To determine the latter we calculated the text prevalence of the player relations in the knowledge graph (see Sect. 3.2). As Table 3 shows, the most prevalent (and thus useful for

Table 1. Metric values and actions

Metric value	Diagnosis	Action
High Lexical Ambiguity	The NER system cannot perform well enough Word Sense Disambiguation	Improve the linguistic analysis component of the NER system
High Global Ambiguity	Many of the input texts are not really related to the domain of the target entities	Use a domain/topic classifier in order to filter out the non-relevant texts and apply the NER process only to the relevant ones
High KGA_1 and low KGA_2	The evidence knowledge graph may contain several non-target entities that hamper the disambiguation process rather than helping it	Prune the evidence knowledge graph in order to remove non-essential, noisy entities
Low Knowledge Graph Richness	Knowledge Graph is not adequate as disambiguation evidence	Enrich the knowledge graph starting from the most prevalent relations
High Knowledge Graph Richness but low Text Prevalence	Knowledge Graph is not adequate as disambiguation evidence	Change or expand the knowledge graph with entities that are more likely to appear in the texts
Low Knowledge Graph Text Prevalence and Low Target Entity Ambiguity and Knowledge Graph Ambiguity	The system's minimum evidence threshold is too high	Decrease the threshold

disambiguation) relations were those between players and the their current team, current co-players and current managers; so we kept those and discarded the rest.

Then we applied again Knowledge Tagger but with the pruned knowledge graph and this time precision and recall were found to be 82 % and 80 % respectively. Thus, our framework managed to provide a fairly accurate diagnosis for the initially mediocre effectiveness of our NER system in the particular case (i.e., that the knowledge graph was bigger than needed) and point us to an action (the pruning of the graph) that actually increased this effectiveness.

4.2 Case 2: Resolving Companies in News Articles

In this case our task was to detect and disambiguate mentions of technology startups within news articles coming from a variety of news sources (newspapers, blogs, specialized websites like techcrunch etc.). For that, we had at our disposal a thesaurus of 4000 company entities as well as a custom-built knowledge graph that contained useful knowledge about each company like its founders, investors, competitors and business areas. Running Knowledge Tagger with this knowledge graph as disambiguation evidence gave us a precision of 35 % and a recall of 50 %, both of which of course were rather low.

To identify the underlying reasons for this low effectiveness, we applied again our diagnostic framework, starting with the identification of the ambiguity types

Table 2. Ambiguity metric values for football case

Metric	Value
Lexical Ambiguity	1 %
Target Entity Ambiguity	30 %
KGA_1	56 %
KGA_2	4 %
Global Ambiguity	2 %

Table 3. Text prevalence of knowledge graph relations and relation paths in the football case

Relation	Prevalence
Relation between soccer players and their current club	85 %
Relation path between players and their current co-players	95 %
Relation path between players and their current managers	75 %
Relation between players and their nationality	10 %
Relation between players and their place of birth	2 %
Relation between players and their spouse	0 %

we were up against. As Table 4 shows, contrary to the football case, our main problem in this scenario was not the ambiguity between startups and/or other related entities in the knowledge graph but the global ambiguity, i.e., the ambiguity between startups and entities outside our domain. A posteriori, this was somewhat expected as the news we were analyzing were not necessarily related to startups or technology. Moreover, there was a considerable lexical ambiguity as several companies had names like "Factual", "Collective" and "Prime".

Given the high global ambiguity, we built and applied, as suggested by our framework, a simple binary classifier to filter out news articles that were not related to our domain. The classifier was based on the multinomial Naive Bayes algorithm and was trained on a set of 400 news articles (200 within the domain and 200 outside), achieving an accuracy of 90 %. Running Knowledge Tagger only on the classified as domain-specific news articles resulted in a substantially increased precision of 72 % while recall stayed roughly the same (52 %). At the same time, in order to deal with the considerable lexical ambiguity, we incorporated to the linguistic analysis component of our system (which is based on GATE) some heuristic rules like, for example, the rule that text terms that refer to startups should start with a capital letter. This increased precision to 78 % and recall to 57 %.

To see if any more improvements were possible, we measured the knowledge graph's prevalence in the texts which turned out to be low. In fact, almost 40 % of the texts contained no evidential entities at all while most of the graph's relations had small prevalence (see Table 5). Based on this fact and the low

Table 4. Ambiguity metric values for companies case

Metric	Value
Lexical Ambiguity	10 %
Target Entity Ambiguity	4 %
KGA_1	4 %
KGA_2	3 %
Global Ambiguity	40 %

Table 5. Text prevalence of knowledge graph relations and relation paths in the companies case

Relation	Prevalence
Relation between companies and the business areas they are active in	50 %
Relation between companies and their founders	40 %
Relation between companies and their competitors	35 %
Relation between companies and their CEO	20 %
Relation between companies and their investors	15 %
Relation between companies and their CFO or CMO	6 %

scores for Target Entity and Knowledge Graph Ambiguity, we ran Knowledge Tagger again but with a reduced minimum evidence threshold; this increased recall to 62 %. Thus, again, our diagnostic framework proved quite useful in determining the underlying causes of our NER's ineffectiveness and guiding us to the appropriate remedial actions.

5 Discussion

The framework we have presented in this paper has been derived from the experiences we had in building NER solutions for actual commercial clients and the two application cases (and their examples/datasets) that we have described reflect exactly those experiences. That is why the quantitative results we report are from our own NER system rather than other systems. These cases may not be covering all possible situations, but they do illustrate how different two NER scenarios may be.

Regarding the level of automation, the framework is applied as follows: First, a diagnostic set of texts is manually created. Second, all metrics are automatically calculated and shown to the user. Third, the metrics are manually interpreted by the user using the guidelines of Sect. 3.3. Thus, our currently implemented system supports the calculation and (basic) visualization of the metrics. The automation of the metrics interpretation is left as future work as it requires a more formal definition of both the metrics and the diagnostic rules (e.g., the definition of "lower than expected" or the comparison of the different metrics).

Given that, the main effort one needs to invest in applying the framework involves the manual annotation of the texts and the execution of the remedial actions the metrics will suggest (e.g., the graph's pruning). In the first use case we described in this paper (football), the application of our framework took us about a week as the texts were short, the domain rather small (Spanish league only) and the pruning of the graph easily done by a few SPARQL queries. The second case (startups) was more demanding (around 3 weeks) as we had to deal with longer texts and enhance Knowledge Tagger with better word sense disambiguation and domain filtering capabilities. Of course, these estimations cannot be considered as any kind of benchmark.

A key insight that one can derive from our framework regarding NER effectiveness is that evidential knowledge should not be applied in a blind manner; in some cases more knowledge may be required (see knowledge graph expansion/enrichment actions in Table 1) but in other cases less knowledge is actually better (see knowledge graph pruning actions in Table 1 as well as the case in Sect. 4.1). In other words, it's not so much the amount of knowledge that counts but its appropriateness to the particular scenario. Our metrics facilitate the assessment of this appropriateness and thus the selection of the optimal knowledge. A second insight is that it's not always necessary to have the optimal evidential knowledge in order to get a satisfactory effectiveness; as the second case in Sect. 4.2 showed, domain filtering and better lexical matching rules were enough to increase NER precision to an acceptable level. Again, the framework's metrics are crucial in recognizing such situations.

Of course, it has to be noted that our framework is rather informal and not necessarily applicable to all NER systems. It is based on insights we have extracted from studying and using existing NER systems (including our own) in real-world scenarios and it's primarily targeted to practitioners that do not have necessarily deep knowledge of NER algorithms and theoretical frameworks, but still need to have some control over their systems' performance.

6 Conclusions and Future Work

In this In-Use paper we have considered the task of Named Entity Resolution and we have defined a Diagnostics Framework for troubleshooting and optimizing corresponding systems in industrial scenarios. Our motivation for this work has been the empirical fact that a NER system's satisfactory performance in a given scenario does not constitute a trustworthy predictor of its performance in different settings. As industrial clients typically expect a high and consistent performance from the NER solutions they pay for, our framework helps NER system developers and consultants identify the reasons why their system performs unsatisfactorily in a given scenario and take appropriate actions to increase performance.

In defining our framework we have first identified the main factors that affect NER effectiveness; two of these are (i) the level of ambiguity that characterizes the scenario's entities and (ii) the adequacy of the contextual evidence applied

for disambiguation. Then we have defined metrics and processes for quantifying these factors and we have linked the values of these metrics to specific actions (see Table 1) that, as Sect. 4 shows, are able to increase NER effectiveness.

As the Diagnostics Framework is currently implemented as part of our own NER system (Knowledge Tagger), our immediate future work will focus on implementing it in a more generic way so that different systems could make use of it. A key feature of such an implementation will be the comprehensive and intuitive visualization of the metrics so that the framework's users can easily interpret their values. Moreover, we intend to extend the framework with metrics for measuring the evidential adequacy of textual knowledge resources as well as any other metrics that we may find useful. Finally, in the longer term, we intend to investigate whether and in what way could this metric-based optimization of NER systems be performed fully automatically, i.e., having the system itself rather than human users interpret the metrics and take up the corresponding actions.

References

1. Alexopoulos, P., Villazon-Terrazas, B., Gomez-Perez, J.M.: Knowledge tagger: customizable semantic entity resolution using ontological evidence. In: Lohmann, S. (ed.) I-SEMANTICS (Posters & Demos). CEUR Workshop Proceedings, vol. 1026, pp. 16–19. CEUR-WS.org (2013)
2. Bos, J.: A survey of computational semantics: representation, inference and knowledge in wide-coverage text understanding. Lang. Linguist. Compass **5**(6), 336–366 (2011)
3. Ferragina, P., Scaiella, U.: 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, CIKM 2010, pp. 1625–1628. ACM, New York (2010)
4. Gangemi, A.: A comparison of knowledge extraction tools for the semantic web. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 351–366. Springer, Heidelberg (2013)
5. Hassell, J., Aleman-Meza, B., Arpinar, I.B.: Ontology-driven automatic entity disambiguation in unstructured text. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 44–57. Springer, Heidelberg (2006)
6. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 782–792. Association for Computational Linguistics, Stroudsburg (2011)
7. Kemmerer, S., Grossmann, B., Müller, C., Adolphs, P., Ehrig, H.: The neofonie NERD system at the ERD challenge 2014. In: Proceedings of the First International Workshop on Entity Recognition, ERD 2014, pp. 83–88. ACM, New York (2014)
8. Kleb, J., Abecker, A.: Entity reference resolution via spreading activation on RDF-graphs. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 152–166. Springer, Heidelberg (2010)

9. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 457–466. ACM, New York (2009)
10. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, I-Semantics 2011, pp. 1–8. ACM, New York (2011)
11. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Lang. Cogn. Process. **6**(1), 1–28 (1991)
12. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 509–518. ACM, New York (2008)
13. Navigli, R.: Word sense disambiguation: a survey. ACM Comput. Surv. **41**(2), 10:1–10:69 (2009)
14. Rizzo, G., Troncy, R.: NERD: a framework for evaluating named entity recognition tools in the Web of data. In ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, 23–27 October 2011
15. Usbeck, R., Ngonga Ngomo, A.-C., Röder, M., Gerber, D., Coelho, S.A., Auer, S., Both, A.: AGDISTIS - graph-based disambiguation of named entities using linked data. In: Mika, P., et al. (eds.) ISWC 2014, Part I. LNCS, vol. 8796, pp. 457–471. Springer, Heidelberg (2014)