The role of knowledge in determining identity of long-tail entities

Ilievski, Filip; Hovy, Eduard; Vossen, Piek; Schlobach, Stefan; Xie, Qizhe

published in
Journal of Web Semantics
2020

DOI (link to publisher)
10.1016/j.websem.2020.100565

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

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

The role of knowledge in determining identity of long-tail entities

Filip Ilievski [a,b,*], Eduard Hovy [c], Piek Vossen [a], Stefan Schlobach [a], Qizhe Xie [c]

[a] Vrije Universiteit Amsterdam, de Boelelaan 1081 HV, Amsterdam, The Netherlands
[b] Information Sciences Institute, University of Southern California, Marina del Rey, CA, USA
[c] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Received 12 October 2019
Received in revised form 24 January 2020
Accepted 15 April 2020
Available online 25 April 2020

Keywords:
Long-tail entities
NIL clustering
Knowledge-based completion

ABSTRACT

Identifying entities in text is an important step of semantic analysis. Some entity mentions comprise a name or description, but many include no information that identifies them in the system's knowledge resources, which means that their identity cannot be established through traditional disambiguation. Consequently, such NIL (not in lexicon) entities have received little attention in entity linking systems and tasks so far. However, given the non-redundancy of knowledge on NIL entities, their lack of frequency priors, their potentially extreme ambiguity, and their numerousness, they constitute an important class of long-tail entities and pose a great challenge for state-of-the-art systems. In this paper, we describe a method for imputing identifying knowledge to NILs from generalized characteristics. We enrich the locally extracted information with profile models that rely on background knowledge in Wikidata. We describe and implement two profiling machines using state-of-the-art neural models. We evaluate their intrinsic behavior and their impact on the task of determining the identity of NIL entities.

1. Introduction

Knowledge scarcity is (unfortunately) a rather prevalent phenomenon, with most instances being part of the Zipfian long tail. Applications in Information Extraction (IE) suffer from **hunger for knowledge**, i.e., a lack of information on the tail instances in knowledge bases (KBs) and in communication. Knowledge missing during IE system processing is traditionally injected from KBs, which attempt to mimic the background knowledge possessed and applied by humans. However, current KBs are notoriously sparse [1], especially on long-tail instances. Not only there are many instances with scarce knowledge, but most (real or imaginary) entities in our world have no accessible representation at all. This poses a limitation to the Entity Linking (EL) task, as we are unable to determine the identity of the vast majority of entities through traditional disambiguation.[1]

Within EL, the forms that refer to non-represented entities are simply resolved with a reference to a 'NIL' (not in lexicon). Taking a step further, the TAC-KBP NIL clustering task [2,3] requires the forms which refer to the same NIL entity within a dataset to be clustered together. We note, however, that the utility of this clustering is limited to the current dataset — provided that no additional information is stored about this entity, it is simply not possible to distinguish it from any other NIL entity found in another dataset, nor to link any form outside of this dataset to that instance, as it contains no description whatsoever. Addressing these limitations and bridging the gap between NIL clustering and EL, here we create formal descriptions for NIL entities that can easily be stored and linked to in the future.

In a real setting, NIL clustering quickly becomes very complex. Consider a document D_1 that mentions a NIL entity with a name 'John Smith'. Given a mention of the same form in another document D_2, we need to decide whether the two documents report on the same person, or a different one sharing the name. Thousands such documents may exist, most of which describe a different person, but some of which could still be about the same person as D_1. Moreover, there is no guarantee that the information about this entity will be consistent across documents.

The lack of frequency priors among these instances (I), the scarcity and non-redundancy of knowledge (II), and the unknown, but potentially extreme, ambiguity (III) make the NIL entities a special case of long-tail entities and a great challenge for IE systems [4]. Considering these factors, the knowledge about the NIL entities needs to be carefully extracted (with both high precision and recall), combined, and stored, in order to allow further reasoning.

In this paper, we investigate **the role of knowledge when establishing the identity of NIL entities mentioned in text**. We

* Corresponding author.
E-mail addresses: ilievski@isi.edu (F. Ilievski), hovy@cmu.edu (E. Hovy), piek.vossen@vu.nl (P. Vossen), k.s.schlobach@vu.nl (S. Schlobach), qzxie@cs.cmu.edu (Q. Xie).

[1] In this work, Entity Linking refers to the task of mapping already recognized entity mentions in text to their representation in a knowledge base.

https://doi.org/10.1016/j.websem.2020.100565
1570-8268/© 2020 Published by Elsevier B.V.

expect that even with perfect attribute extraction, it is not always trivial to determine the identity of a long-tail entity across documents due to heterogeneity of information: the information about an entity found in different documents is not necessarily identical or even comparable, as these have been written independently of each other and might exhibit different framing that is relevant in their context.

When two entity descriptions are not directly comparable, how can one decide on their (non-)identity? For instance, if John Smith in D_1 is a Texas-based truck driver and the one in D_2 is a recent Harvard graduate, are they likely to be identical? Although these two descriptions do not reinforce or contradict each other in a logical sense, people immediately make a judgment that allows them to fill such knowledge gaps with ease. These judgments are based on knowledge about associations among facet values and a continuously evolving collection of cognitive expectations and stereotypical profiles [5].[2] People assume that they are entitled to fill knowledge gaps with their expectations unless contradictory evidence is explicitly presented.

Growing amounts of data and the advent of workable neural (deep) models raise the natural question: how can one build models that capture such prominent cognitive skills of people? How can one fill knowledge gaps when these cannot be distilled directly from communication nor retrieved from existing knowledge bases? A popular computational task of completing missing values in a knowledge graph is Knowledge Base Completion (KBC), where the system is asked to add new concrete facts given other instantial information. We motivate a variant of KBC called **profiling**, where we predict expectations over value classes rather than predicting specific values with high precision. These expectations should of course be conditioned on whatever (partial) information is provided for any test case, and automatically adjusted when any additional information is provided. We see profiling as a common and potentially more helpful capability for filling gaps in IE, where knowing preferences or ranges of expectations (rather than concrete values) is often necessary in order to perform reasoning and to direct interpretation on underspecified (long-tail) data.

We design and implement two profiling machines, i.e., background knowledge models that fill knowledge gaps, based on state-of-the-art neural methods. The profiling machines are evaluated extrinsically on a NIL clustering task, where the evaluation data concerns long-tail event participants (of type *Person*) from the gun violence domain. To enable this evaluation, we match the local, incomplete context extracted from text to its corresponding profile. We investigate the effect of profiling on top of varying amount and quality of information extracted from text. The robustness of our models is tested by a systematic increase in the ambiguity of forms. By combining explicit and implicit knowledge, we are able to gain insight into the role of background knowledge models when establishing identity of long-tail entities. In addition, we perform an intrinsic evaluation of the behavior and performance of the profiling models.

We summarize the contributions of this paper as follows:

1. We formulate a set of hypotheses about the role of background knowledge models in establishing identity of long-tail entities (Section 3).
2. We motivate a KBC-variant of profiling (Section 4) to support the interpretation of contextual, long-tail instances in IE. In profiling, we predict expectations over value classes rather than predicting specific values with high precision.
3. We describe two generic state-of-the-art neural methods that can be easily instantiated into profiling machines for generation of expectations (Section 4).
4. We illustrate the usefulness of the profiling methods *extrinsically* on a task of NIL clustering (Section 6). By determining identity of entities without popularity or frequency prior and under very high ambiguity, we deliberately address the true challenges for the entities in the distributional 'tail'.
5. We evaluate the profiling methods *intrinsically* against known values in Wikidata and against crowd judgments, and compare the results to gain insight in the nature of knowledge captured by profiles (Section 7).
6. All code and data is made available to facilitate future research.

2. Related work

The task of determining identity of NIL entities has been motivated by NIL clustering, a recent addition to the standard task of EL. We review previous approaches for EL and NIL clustering in Section 2.1. A knowledge-based reasoning over identity in text resembles previous work on attribute extraction and slot filling (Section 2.2). Existing tasks and typical state-of-the-art approaches for filling knowledge gaps are covered in Sections 2.3 and 2.4.

2.1. Entity linking and NIL clustering

Entity Linking. EL facilitates the integration and reuse of knowledge from existing KBs into corpora, established through a link from an entity mention in text to its KB representation. Many EL systems have been proposed in the past decade [6–11]. These systems typically rely on various probabilistic algorithms for graph optimization or machine learning, in order to pick the correct entity candidate for a surface form in a given context.

Most recent approaches for entity linking rely on entity and word embeddings, trained over Wikipedia or its structured derivatives. [12,13] show how to compute embeddings separately on a knowledge graph (e.g., DBpedia) and on a collection of documents (e.g., Wikipedia), and how to align both (e.g., by using information from the hyperlinks in Wikipedia). These methods arguably work well for unknown entities at test time. [14,15] apply such pre-trained embeddings on the task of Entity Linking. At test time, they compute an embedding of an entity mention and look for the most similar entity embedding from the KB, using a metric like cosine similarity. Notably, these works exclude NIL entities from their evaluation. Another option is to train the entity and word embeddings jointly over Wikipedia documents only, with no alignment function needed [16]. The pre-trained embeddings of each candidate are then compared (using cosine similarity) with the embedding of the mention that is being resolved. This similarity, enriched with other features about that candidate and the textual content of a mention, is used as features in a neural network that decides whether a candidate fits the textual context, or not.

While state-of-the-art systems exhibit sophisticated architectures and report high results, it has been argued and shown that these base their performance on the frequent and unambiguous 'head' cases, while performance drops significantly when moving towards the rare and ambiguous 'long-tail' entities [4,17,18]. The analysis in [19] demonstrates that state-of-the-art EL systems lack much human-like knowledge and argues that this causes the lower performance on long-tail entities, where this knowledge is most needed. Similarly, Sakor et al. [11] introduce the role of background knowledge in targeting entity linking over short text.

[2] We consider the terms 'attribute', 'property', and 'facet' to be synonymous in the context of this work. Hence, we will use them interchangeably.

An extreme category of long-tail entities in the EL task are the NIL entities. **NIL entities** are entities without a representation in a knowledge base [2]. These are typically considered to have low frequencies within a corpus and/or to be domain-specific. Esquivel et al. [18] report that around 50% of the people mentioned in news articles are not represented in Wikipedia. Since Wikipedia and its structured data counterparts are almost exclusively used as an anchor in EL, this means that for half of all people mentions, the EL task is nonsensical.

NIL clustering. The Text Analysis Conference's Knowledge Base Population (TAC-KBP) [2,3] challenge has introduced a task of NIL clustering, asking systems to cluster NIL mentions that are coreferential. Although state-of-the-art EL systems do not perform NIL clustering or do not detail their clustering algorithm, several NIL clustering approaches were proposed within the TAC competitions. In [20], three baseline techniques to cluster NIL entities are proposed: 1. 'term' (cluster terms that share the same mention) 2. 'coref' (cluster based on in-document coreference) 3. 'KB' (use information from a KB where this NIL actually does exist) . Graus et al. [21] first translate the full documents to TF.IDF weighted vectors, then perform hierarchical agglomerative clustering algorithm on the vectors of all documents that were previously labeled as NIL by the system. [22] performs NIL clustering together with EL, by first clustering mentions based on their term and certain features: type, verbal context, etc., and then optionally linking the cluster to a KB entity. When resolving a nominal mention, the most effective approach is to apply within-document coreference resolution to resolve it to a name mention [3]. Linking each identified nominal mention to its closest person name mention can yield 67% accuracy [23].

Notably, NIL clustering is a fairly marginal part of the full task of interpretation of entity mentions. The clustering of NILs is either not done at all, or not reported/assumed to be done in a default (perhaps term-based) manner. Reported methods are typically based on the term itself, coreference, and verbal context. We focus on establishing identity of NIL entities, and we apply a method that is based on background knowledge models and reasoning over entity attributes.

2.2. Attribute extraction

Previous work on attribute extraction in the field of Information Retrieval (IR) [24,25] resembles our task and method in several aspects: 1. multiple documents may point to the same person, and there is ambiguity of person names; 2. many entities belong to the long tail; 3. the method is to represent people with attributes that are extracted from text; 4. modeling of instances is based on a restricted set of attributes. However, given that the goal is to find personal websites in a search engine, the context of disambiguation is poorer. Moreover, in most cases, clustering on a document level is sufficient, since there is a single central entity per document.

The goal in the slot filling task is use a document collection to fill in values for specific attributes ('slots') of given entities. The per-attribute F1-scores show large variance between attributes, between systems, and across datasets. Angeli et al. [26] report an average slot filling F1-score of 37%, the average score in [27] is 53%, whereas the average F1-scores in the TAC KBP slot filling competitions in 2016 and 2017 for English are between 10% and 25%. Considering this instability in performance and the fact that some of the properties we use were customized to fit the information found in a document, we opted to build our own lexical attribute extractors (described in Section 5.4), whose performance is comparable to that reported in the aforementioned papers.

2.3. Knowledge base completion (KBC)

Most facet values in both Freebase and Wikidata are missing [1]. KBC adds new facts to knowledge bases/graphs based on existing ones. Two related tasks are link prediction (with the goal to predict the subject or object of a given triplet, usually within the top 10 results) and triplet completion (a binary task judging whether a given triplet is correct).

In the past decade, KBC predominantly employed deep embedding-based approaches, which can be roughly divided into tensor factorization and neural network models [28]. TransE [29] and ITransF [30] are examples of neural approaches that model the entities and relations in embedding spaces, and use vector projections across planes to complete missing knowledge. Tensor factorization methods like [31] regard a knowledge graph as a three-way adjacency tensor. Most similar to our work, Neural Tensor Networks [32] also: 1. aim to fill missing values to mimic people's knowledge; 2. evaluate on structured relations about people; 3. rely on embeddings to abstract from the actual people to profile information.

Universal Schema [33] includes relations extracted from text automatically, which results in larger, but less reliable, initial set of relations and facts. It was designed for the needs of NLP tasks such as fine-grained entity typing [34].

As apparent from the amount and diversity of work described here, KBC research is a well-established research direction that encapsulates various efforts to complement knowledge about real-world instances with probable new facts. We define profiling as a variant of KBC that aims: 1. to generate an expectation class for *every facet* of a category/group, rather than suggesting missing facts; 2. to provide a *typical distribution* (not a *specific* value) for the attributes of a specific group. These differences make profiling more useful for reasoning over incomplete data in NLP and AI applications, and related to cognitive work on stereotypes.

KnowledgeVault (KV) [1] is a probabilistic KB by Google, which fuses priors about each entity with evidence about it extracted from text. Despite using a different method, the priors in KV serve the same purpose as our profiles: they provide expectations for all unknown properties of an instance, learned from factual data on existing instances. Sadly, the code, the experiments, and the output of this work are not publicly available, thus preventing further analysis of the priors, their relation to cognitive/cultural profiling as done by humans, and their applicability in IE identity tasks.

2.4. Other knowledge completion variants

Several other, quite distinct research areas, are relevant to profiling. We briefly list some of the most relevant work.

Data imputation. Data imputation refers to the procedure of filling in missing values in databases. In its simplest form, this procedure can be performed by mean imputation and hot-deck imputation [35]. Model-based methods are often based on regression techniques or likelihood maximization [36]. These efforts focus on guessing numeric and interval-valued data, which is a shared property with the related task of guesstimation [37]. In contrast, profiling aims to predict typical *classes* of values. Moreover, it is unclear how to apply data imputation in IE use cases.

Estimation of property values. Past work in IR attempted to improve the accuracy of retrieval of long-tail entities by estimating their values based on observed head entities. Most similar to profiling, the method in [38] estimates a property of a long-tail entity based on the community/ies it belongs to, assuming that each entity's property values are shared with others from the same community. Since the goal of this line of research is to improve the accuracy of retrieval, the generalization performed is rather ad hoc, and the knowledge modeling and management aspects have not been investigated in depth. Moreover, the code, the experiments, and the results of this work are not available for comparison or further investigation of its usefulness for IE applications.

Social media analysis. Community discovery and profiling in social media is a task that clusters the online users which belong to the same community, typically using embeddings representation [39], without explicitly filling in (representing) missing property values.

Local models that infer a specific property (e.g., user's location) based on other known information, such as her social network [40] or expressed content [41], also address data sparsity. These models target specific facets and data types, thus they are not generalizable to others. Similarly to models in KBC, they lack cognitive support and fill specific instance values rather than typical ranges of values.

Probabilistic models. Prospect Theory [42] proposes human-inspired cognitive heuristics to improve decision making under uncertainty. The Causal Calculus theory [43] allows one to model causal inter-facet dependencies in Bayesian networks. Due to its cognitive background and assumed inter-facet causality, profiling is a natural task for such established probabilistic theories to be applied and tested.

Stereotypes. Stereotype creation is enabled by profiling. The practical uses of stereotypes are vast and potentially ethically problematic. For example, [44] claims that embedding representations of people carry gender stereotypes; they show that the gender bias can be located in a low-dimensional space, and removed when desired. We leave ethical and prescriptive considerations for other venues, and note that artificially removing certain profiling-relevant signals from the data makes embeddings far less useful for downstream IE tasks when evaluated against human performance.

3. Task and hypotheses

This section provides an explanation of the NIL clustering task, describes our notion of background knowledge, and lists the set of hypotheses that we will investigate.

3.1. The NIL clustering task

NIL entities are those that do not have a representation in the referent knowledge base K to which we are linking. Given that most world entities are NIL entities, the real-world ambiguity among NIL entities is potentially far larger compared to the ambiguity of the non-NIL entities that constitute the majority of our EL datasets. In addition, the lack of an existing representation of these instances in a knowledge base and the minimal textual information on them, means that there is very little redundancy of information on the NIL entities. The ambiguity and knowledge sparsity of these entities require systems to extract information on NIL mentions carefully and with high precision, but also to be able to have the right expectations about the pieces of information that have been deliberately left out.

NIL clustering Similar to the NIL clustering task introduced in TAC-KBP, the aim in this work is to cluster NIL mentions that are coreferential. Formally, the set of forms $f_{i,1}, \ldots, f_{i,n}$ belongs to the same cluster with the set of forms $f_{j,1} \ldots, f_{j,m}$ if and only if they refer to the same entity instance.

3.2. Types of knowledge

NIL entities have no accessible representation, which prevents one to establish their identity with traditional EL. There is, however, information about these entities in textual documents, which can be used as basis to perform clustering or generate a new representation. Moreover, this knowledge found in text can be enhanced with various background knowledge found in the metadata of the document, other documents, or knowledge bases.

MacLachlan and Reid [45] define four types of contextual knowledge that are essential for humans to interpret text. These are: *intratextual*, *intertextual*, *extratextual*, and *circumtextual* knowledge. In [19], we adapted these four categories to the task of establishing identity of entities in text:

1. *Intratextual knowledge* is any knowledge extracted from the text of a document, concerning entity mentions, other word types (e.g., nouns, verbs), and their order and structure in the document. It relates to framing new and given information and notions such as topic and focus. Central entities in the discourse are referred to differently than peripheral ones.
2. *Extratextual knowledge* concerns any entity-oriented knowledge, found outside the document in (un)structured knowledge bases. It can be episodic (instantial) or conceptual. The former is the knowledge about a concrete entity: its labels, relation to other entities, and other facts or experiences. Conceptual knowledge refers to the expectations and knowledge gaps that are filled by an abstract model (i.e., ontology), representing relations between types of entities.
3. *Circumtextual knowledge* refers to the circumstances through which text as an artifact has come into existence. Documents are published at a specific time and location, written by a specific author, released by a certain publisher, and potentially belong to some series. These circumstances frame the written text and aid the interpretation of the mentioned entities.
4. Documents are not self-contained and rely on *intertextual (cross-document) knowledge* distilled by the reader from related documents. They are published in a stream of information and news, assuming knowledge about preceding related documents, which typically share the same topic and community, and may be published around the same time and location. Early documents that introduce a topic typically make more explicit reference than those published later on when both the event and the topic have evolved.[3]

In our line of work, the term *background knowledge* refers to the union of the intertextual, circumtextual, and extratextual knowledge. Background knowledge has been shown to be largely excluded by current systems that establish identity of entities in text, which has grave consequences for the long-tail cases [19].

3.3. Research goals and hypotheses

In this paper, we seek to understand the role of various knowledge that is potentially needed in order to establish identity of a NIL entity. It is unclear to which extent the intratextual knowledge, distilled from text, is sufficient to establish identity of long-tail entities. It is also unclear what is the potential of implicit, background knowledge to improve the performance on

[3] Compare the use of hashtags in Twitter streams once an event becomes trending.

Table 1
Hypotheses on the role of profiling for establishing identity of long-tail entities.

ID	Hypothesis	Sec
C1	Assuming that the available information in documents is sufficient, perfect attribute extraction would perform NIL clustering reliably.	6.1
C2	Automatic attribute extraction leads to a decline in the clustering performance compared to perfect extraction.	6.1
C3	Assuming sufficient information and perfect attribute extraction, the role of profiling is minor.	6.2
C4	Profiling can improve clustering performance when attribute extraction is less accurate.	6.2
C5	The overall clustering performance is inversely proportional with ambiguity.	6.3
C6	The effect of profiling is larger when the ambiguity is higher.	6.3

Table 2
Hypotheses on the behavior of profiling.

ID	Hypothesis	Sec
P1	Profiling corresponds to the factual instance data.	7.1
P2	Profiling corresponds to human expectations.	7.2
P3	Profiling is more helpful when the entropy is higher.	7.1, 7.2
P4	Profiling is more helpful for larger value spaces.	7.1
P5	Profiling performs better with more known facets.	7.1

this task, and what is the sensitivity of the performance to varying degrees of data ambiguity.

Even though background knowledge is intuitively essential for better information extraction on long-tail cases and further inclusion has been argued for (cf. [19]), to the best of our knowledge, no previous work has investigated this quantitatively in a systematic manner. In order to gain insight into our questions, we put forward six hypotheses about the role of various kinds of knowledge in the task of NIL clustering. These hypotheses are listed in Table 1. On the one hand, we investigate the impact of intratextual knowledge, distilled from information that is explicitly stated in text. On the other hand, we capture implicit expectations by people when processing such a document by including profiling models that generalize over existing background knowledge.[4] Our current profiling machines are based on extratextual knowledge; integrating and investigating circumtextual and intertextual knowledge is planned for future work. Profiling as a cognitive task and our computational variant inspired by it, are introduced in detail in the next section of this paper.

Since the profiling components are central to our work, we perform *intrinsic analysis* of their behavior, by comparing them to human judgments and existing instances in Wikidata. Our expectations for the intrinsic evaluations are summarized in Table 2. These resemble studies on stereotyping accuracy in social psychology (cf. [46]), as well as to previous analyses that measure the sensitivity of knowledge graph representation systems with respect to sparsity, reliability, and diversity of knowledge (cf. [47]). Nevertheless, we expect that these hypotheses reveal new insights, considering that profiling is a novel variant of KBC designed to fill knowledge gaps in text.

4. Profiling

4.1. Aspects of profiles

Following [48] we define a profile as a set of beliefs about the attributes of members of some group. A stereotypical profile is a type of *schema*, an organized knowledge structure that is built from experience and carries predictive relations, thus providing a theory about how the social world operates [49]. As a *fast* cognitive process, profiling gives basis for acting in uncertain/unforeseen circumstances [50]. Profiles are "shortcuts to thinking", that provide people with rich and distinctive information about unseen individuals. Moreover, they are *efficient*, thus preserving our cognitive resources for more pressing concerns.

Profile accuracy reflects the extent to which beliefs about groups correspond to the actual characteristics of those groups [46]. Consensual ones have been empirically shown to be highly accurate, especially the demographic (race, gender, ethnicity, etc.) and other societal profiles (like occupations or education), and somewhat less the political ones [46]. This high accuracy does not mean that profiles will correctly describe any group individual; they are a statistical phenomenon. Thus, the findings in [46] that most profiles are justified empirically are of great importance for AI machines: it means that they can be reliably inferred from individual facts, which (unlike many profiles themselves) are readily available.

Profiles exist at various levels of specificity for facets and their combinations. A profile of 20th century French people differs from a profile of 20th century people in general, with more specificity in what kind of food they eat and what movies they watch, or from the profile of French people across all ages. Added information usually causes the initial expectations to change ("shift"), gradually narrowing the denotation group in a transition toward ultimately an individual. The shift may be to a more accurate profile (what in [51] is called an "accurately shifted item"), or the opposite ("inaccurately shifted item").

4.2. Examples

People have no trouble making and using profiles/defaults:

P1 is male and his native language is Inuktitut. What are his citizenship, political party, and religion? Would knowing that he was born in the 19th century change one's guesses?

P2 is a member of the American Senate. Where did he get his degree, and what is his native language?

P3 is an army general based in London. What is P3's stereotypical gender and nationality? If P3 gets an award "Dame Commander of the Order of the British Empire", which expectations would change?

Presumable answers to these questions are as follows. P1 is a citizen of Canada, votes for the Conservative Party of Canada, and is Catholic. However, P1's 19th century equivalent belongs to a different party. P2 speaks English as main language and graduated at Harvard or Yale University. Finally, the initial expectation on P3 of a male Englishman switches to a female after the award evidence is revealed.

Most of us would agree with the suggested answers. Why!? What is it about the Inuktitut language that associates it to the Conservative Party? Why is the expectation about the same person in different time periods different? Why does the sole change of political party change the expectation of one's work position or cause of death? Despite the vast set of possibilities, these kinds of knowledge gaps are easily filled by people based on knowledge about associations among facet values, and give rise to a continuously evolving and changing collection of cognitive expectations and stereotypical profiles.

IE systems require such human-like expectations in order to deal with knowledge sparsity and the ambiguity of language. In this paper, we illustrate this on a task of establishing identity of long-tail entities.

[4] By 'implicit expectations', we mean probabilistic background knowledge that is not explicitly stated in text, yet it is commonly used by people to interpret text.

4.3. Definition of a profile

An ideal representation of a long-tail entity would entail combining explicitly stated information in text with implicit expectations based on background knowledge. To illustrate this, let us consider our John Smith from D_1 (Section 1): a Texas-based truck driver. Our implicit knowledge would enhance this information with expectations about his native language (almost certainly English), religion (e.g., 98% chance of being Christian, 2% others), and gender (e.g., 85% male, 15% female). Here knowledge plays a role both in the local, textual context, as well as in the enrichment of this local context with expectations stemming from a profile. The hunger for knowledge on the long-tail entities can best be bridged by rich knowledge on both sides of the puzzle.

We use property-value pairs to represent the set of facts in the local context, which are based on intratextual knowledge found in text. The profiles are formalized as learned probability distributions over the properties with no known value, and rely exclusively on extratextual knowledge.[5]

For a set of identical surface forms f mentioned in text, we define its locally learned description, i.e., **local context** *lc(f)*, as a set of property-value pairs extracted from text:

$$lc(f) = \{(p_i, v_{ij}) | p_i \in P \wedge v_{ij} \in V_i\}$$

Here P is the set of the considered properties, and V_i is the domain of values for a property p_i.

Local contexts may not be sufficient to establish identity of entity mentions because: 1. The same form can refer to different local contexts, which might or might not be identical (ambiguity) 2. Different forms sometimes refer to the same local context (variance). Then our task is to establish identity between some, but not all, local contexts. This is non-trivial, since the local concepts are often not directly comparable. For example, while $lc_1 = \{(\text{'}Birthplace\text{'}, \text{'}Mexico\text{'})\}$ and $lc_2 = \{(\text{'}Birthplace\text{'}, \text{'}NewYork\text{'})\}$ are certain to be non-identical, none of them can be directly compared to $lc_3 = (\text{'}Ethnicity\text{'}, \text{'}Hispanic/Latin\text{'})$.

For this purpose, we introduce the notion of **profiles**: globally learned descriptions that help to distinguish local contexts. Given a set P of properties $p_1, \ldots, p_N$ and a local context $lc(f) = (p_1, v_{1k}), \ldots, (p_i, v_{ik})$, we define its profile as a distribution of expected values for the remaining $(N - i)$ properties, namely:

$$profile(f, P, V, T, K) = lc(f) \cup \{(p_{i+1}, d_{i+1}), \ldots, (p_N, d_N)\}$$

where $d_{i+1}, \ldots, d_N$ are distributions of expected values for the properties $p_{i+1}, \ldots, p_N$ given the known property-value pairs in the local concept $lc(f)$. Besides the form and its local context, the *profile* of a form depends on: a set of properties P, their corresponding domain of values V, the set of textual documents T, and the background knowledge K. For a local entity context extracted from text, the goal of profiling is to produce an optimal profile. A profile is considered optimal when its property-value pairs have the highest probability given the background knowledge used for training.

Such global profiles would provide us with a global network to compare and disambiguate local contexts that are not directly comparable. By doing so, we expect that we can cluster identical local contexts and pull apart non-identical local contexts with a higher accuracy. We consider all local contexts that share the same profile to be identical, constituting an **equivalence class**.

In order to fulfill their function of successfully disambiguating two locally derived contexts from text, the set of properties and values constituting a profile needs to fulfill several criteria:

1. to be generic enough[6]
2. to be restricted by what can be learned from background knowledge
3. to be based on what can be found in text
4. to be the minimal set of such properties

These criteria are application-dependent: while in IE tasks they are driven by the information found in text, an IR application could employ criteria like demand/usage [52].

4.4. Neural methods for profiling

Next, we describe neural methods for profiling at scale, and baselines for comparison. The implementation code of these architectures and the code to prepare data to train them are available on GitHub: https://github.com/cltl/Profiling.

4.4.1. Autoencoder (AE)

An autoencoder is a neural network that is trained to copy its input to its output [53]. Our AE contains a single densely connected hidden layer.

Input The input $\boldsymbol{x}$ of the AE is a concatenation of n discrete facet representations $x = x_1, \ldots, x_n$. Each of the facets x_i (e.g., political party) has a vocabulary v_i of possible values which is determined by the training data. We encode an attribute value with an embedding of size N_e, resulting in a size of the embedding input layer of $|\boldsymbol{x}| = n * N_e$. These facet embeddings are initialized randomly and are trained as part of the network.

For example, if the input x represents the entity Angela Merkel (Wikidata ID: $Q567$), it would consist of n embeddings for its n individual attribute values (nationality: German, political party: Christian Democratic Union, etc.).

Output We denote the corresponding output for an input sequence $\boldsymbol{x}$ with $\boldsymbol{z} = g(f(\boldsymbol{x}))$, where f is the encoding and g is the decoding function. The output layer of the AE assigns probabilities to the entire vocabulary for each of the n features. The size of the output layer is a sum over the variable vocabulary sizes of the individual inputs v_i: $|\boldsymbol{z}| = \sum_{i=1}^{n} v_i$.

For instance, if our data consists of three attributes with vocabulary sizes: $v_1 = 300$, $v_2 = 400$, and $v_3 = 500$, then the output layer would concatenate the probabilities for all attribute values, leading to an output layer size $|\boldsymbol{z}|$ of 1,200.

Loss function The AE aims to maximize the probability of the correct class for each feature given inputs $\boldsymbol{x}$, i.e., it is trained to minimize the cross-entropy loss L that measures the discrepancy between the output and the input sequence:

$$L(\boldsymbol{x}, \boldsymbol{z}) = -\sum_{i=1}^{n} [x_i log z_i + (1 - x_i) log(1 - z_i)]$$

We note that we evaluate the correctness of the predicted class against the 'true' one for each attribute separately, while masking the corresponding value in the input. The informativeness of the learning task is further enhanced by a probabilistic dropout that masks part on the input, as well as the fact that the hidden layer in an autoencoder acts as a form of compression.

Sparsity Due to the sparse input, it is crucial that AE can handle missing values. We aid this in two ways: 1. we deterministically mask all non-existing input values; 2. we apply a probabilistic dropout on the input layer, i.e., in each iteration we randomly remove a subset of the inputs (any existing input is dropped with a probability p) and train the model to still predict

[5] Generally speaking, the local context and the profiles could be represented in a variety of ways, and could be based on different kinds of knowledge. Alternative representations (e.g., word embeddings) and an integration of additional knowledge (e.g., circumtextual) would be relevant to investigate in future research.

[6] By 'generic', we mean that the properties and their values should be easily applicable to unseen instances, but also to be meaningful to define their identity. The values for each property, moreover, should be disjoint and easily distinguishable.

these correctly. Although we apply the dropout method to the input layer rather than the hidden layer, we share the motivation with [54] to make the model robust and able to capture subtle dependencies between the inputs. Such dropout helps the AE reconstruct missing data.

4.4.2. Embedding-based neural predictor (EMB)

In our second architecture each input is a single embedding vector $\mathbf{e}$ rather than a concatenation of n facet embeddings. For example, the input for the entity Angela Merkel is its fully-trained entity embedding. The size of the input x is the size of that entity embedding: $|\boldsymbol{e}| = N_e$. In the current version of EMB we use pre-trained embeddings as inputs, and we do not tune their values further. Future work can investigate the benefits of further training/tuning.[7]. We refer the reader to Section 4.4.4 for details on the pre-trained embeddings we use and their training procedure.

We expect that the entity embedding encodes rich information associated with an entity; the task of the neural network EMB is to extract/predict attribute values from it.

Like the AE, EMB has one densely connected hidden layer. For an input $\boldsymbol{x}$ and its embedding representation $\boldsymbol{e}$, the corresponding output is $\boldsymbol{z} = g(h(\boldsymbol{e}))$. The output layer of the embedding-based predictor has the same format as the one of the AE, and the same cross-entropy loss function $L(\boldsymbol{x}, \boldsymbol{z})$. In this architecture, we do not perform masking of the input, as it is not obvious which parts of the input embedding correspond to which attributes.

4.4.3. Baselines

We evaluate the methods against two baselines: a most frequent value (MFV) and a Naive Bayes (NB) baseline.

Most frequent value baseline (MFV) chooses the most frequent value in the training data for each attribute. For instance, since 14.26% of all politicians are based in the USA, MFV's accuracy for profiling politicians' citizenship is 14.26%.

Its motivation is shared with previous frequency-based baselines, such as the most frequent sense (MFS) baseline discussed in [55]. The MFV baseline indicates for which facets and to which extent our methods can learn dependencies that transcend frequency statistics. Given its simplicity, we implement MFV from scratch.

Naive Bayes classifier (NB) [56] applies Bayes' theorem with strong independence assumptions between the features. We represent the inputs for this classifier as one-hot vectors. Naive Bayes classifiers consider the individual contribution of each input to an output class probability. However, the independence assumption prevents it from adequately handling complex inter-feature correlations.

Our NB baseline is based on the `scikit-learn` implementation of the Naive Bayes algorithm.[8]

We deliberately opted for simple baselines with gradually increasing complexity, as these reveal insights into the various complexity levels of the underlying data. Namely, the first baseline would perform well on facets that can be largely predicted based on simple frequency counts. The second baseline goes a step further, as it allows us to model input–output dependencies, but is unable to model inter-feature correlations. We expect that a subset of the predictions can be correctly performed by our simple baselines, whereas many others would require more sophisticated neural techniques such as those described in Sections 4.4.1 and 4.4.2. We study the behavior of our baselines and neural methods in relation to the data properties in Section 7.

Finally, we note that the list of baselines is far from complete and it should be extended with other approaches, including machine learning algorithms like Support Vector Machines (SVM), case-based reasoning algorithms, and state-of-the-art knowledge base completion systems such as the ones reviewed in Section 2.

4.4.4. Model implementation details

We experimented with various parameter values suggested in the literature and opted for the following settings. Both neural models use a single dense hidden layer with 128 neurons. For the AE model, we pick an attribute embedding size of $N_e = 30$. These vectors are initialized randomly and trained as part of the network. We set the dropout probability to $p = 0.5$.

The inputs of EMB are 1000-dimensional vectors representing Freebase entities. These Freebase vectors were trained on a news corpus using the 'word2vec' method, and are publicly available at . In total, this dataset provides us with 1.4 million entity embeddings. They correspond and can be mapped to only a subset of all million Wikidata entities.

Both models were implemented in `Theano` [57]. We used the ADAM [58] optimization algorithm. We train for a maximum of 100 epochs with early stopping after 10 consecutive no-improvement iterations, to select the best model on a held-out validation data. We fix the batch size to 64. When an attribute has no value in an entire minibatch, we apply oversampling: we randomly pick an exemplar that has a value for that attribute from another minibatch and append it to the current one.

5. Experimental setup

Next, we introduce our experimental setup. We start by discussing a modular end-to-end pipeline for NIL clustering, which will allow us to systematically investigate our research question and hypotheses. Then, we present the data we evaluate on, the evaluation metrics we employ to measure the performance of individual components in the pipeline, and the functionality of these components in detail. The code of all experiments can be found on GitHub: https://github.com/cltl/LongTailIdentity.

5.1. End-to-end pipeline

The testing of our hypotheses C1–C6 is enabled by following different paths in a modular end-to-end architecture for NIL clustering. Fig. 1 presents a schematic overview of the components that constitute our end-to-end NIL clustering architecture. The input to the pipeline consists of a set of documents with known entity names (see Section 5.2 for details on the data). The goal of the pipeline is to create clusters of NIL entities. The evaluation of these clusters against the gold clustering output is described in Section 5.3.

The pipeline consists of three main processes: attribute extraction, profiler, and reasoner. The attribute extraction process aims to extract explicitly given information about an entity in a document (we refer the reader to Section 5.4 for more details on our attribute extractors). We experiment with both gold (perfect) attribute extraction, as well as automatic (imperfect) extractors. Once the explicitly mentioned attributes have been extracted, we run profiling models (described in 4.4) to obtain default expectations for the attributes which are not explicitly given. Details on the profiling data and architecture are given in Section 5.5. In the final third step, we perform reasoning in order to cluster the entity mentions in different documents based on the locally extracted information, potentially enriched by the profiling. The reasoners that we used for this step are described in Section 5.6.

[7] As pre-trained embeddings are often not available or easy/quick to create, sometimes such training is unavoidable.

[8] https://scikit-learn.org/stable/modules/naive_bayes.html.

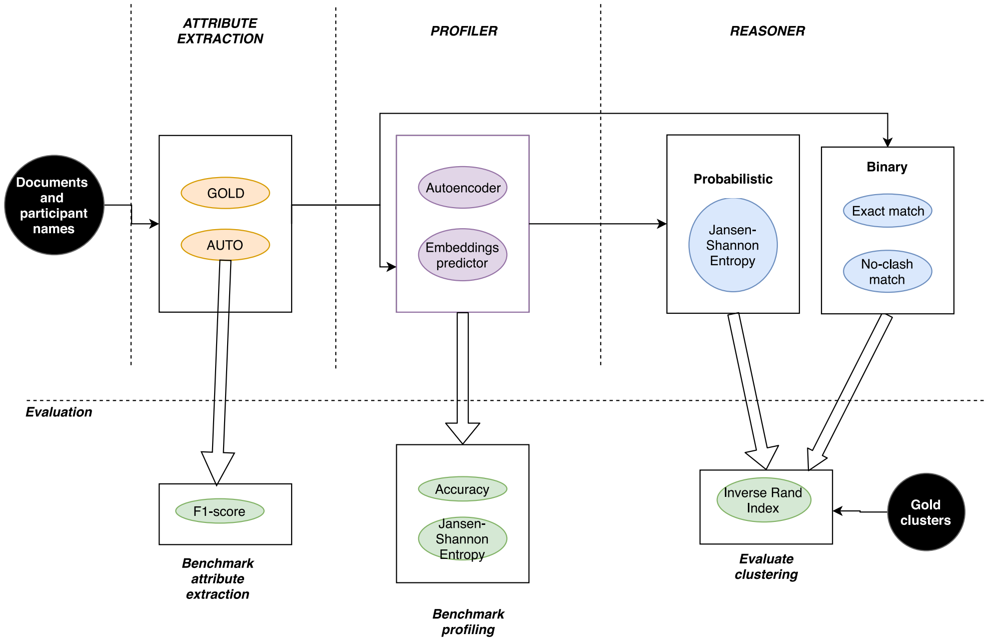

Fig. 1. End-to-end architecture for NIL clustering.

5.2. Data

SemEval-2018 task 5 We test our methods on the Gun Violence domain data from the SemEval-2018 task 5 on 'Counting events and participants in the long tail' [59]. Its data consists of documents which potentially describe the same incident; the goal is then to find the number of incidents that satisfy the question constraints, to decide on the documents that report on these incidents, and count the participants in their corresponding roles. Given that this task evaluates the identity of events, we first need to prepare the data to be suitable for evaluation of the identity of entities.

The SemEval-2018 task 5 data is exceptional in that it describes unknown people participating in local incidents, described in local news documents. As such, these people can safely be assumed to have no representation in the customary knowledge bases, such as DBpedia, but neither in the Linked Open Data cloud. The only way in which we can model the identity of these people is through extensive usage of knowledge, found in the current or other documents, as well as reasoning over conceptual world knowledge.

Each incident comes with structured data describing its location, time, and the age, gender, and name for all participants. From this information, we gather the following properties for each participant: name, gender, age, death year, and death place (the last two attributes apply only if the person was killed).

Data partitions We define two data partitions over this data: 1. **FULL** The entire set of 2,261 incidents used in this task comprises our 'full' data partition. This partition contains consistent annotations on an incident level for the attributes: name, age, gender, death year, and death place, for all participants. We do not know whether these attributes are reported in each of the supporting documents — for the purpose of our experiments we assume this is the case, as the structured data was constructed from the news. 2. **PARTIAL** The partial data consists of 260 incidents described in 457 documents, capturing 472 participants with 456 distinct names.[9] For this subset of incidents, we additionally annotated evidence per document and extracted values for 9 additional attributes. This annotation is described in detail next.

Table 3
Number of incidents, documents, and participants in each of the two partitions.

Partition	# inc	# docs	# participants
Full	2,261	4,479	5,408
Partial	260	457	472

Table 3 presents the number of incidents, documents, and participants in each of the two datasets.

Annotation of explicit values For the partial dataset, we present guidelines.[10] to enrich the properties for each entity with 9 additional properties occasionally mentioned in text, namely: residence, cause of death, past convictions/charges, ethnic group (ethnicity), education level, birth place, native language, political party, and religion. These properties were manually chosen to conform to the requirements listed in Section 4.3 For each property and each person described in the article, we perform two types of annotation based on the information given in text:

1. **Structured (incident-level) annotation** we fill the profile of each entity as much as we can based on the information found in text. For instance, the text: 'Gaby, who only finished high school this summer, is of Chinese origin. ...' provides evidence that the ethnic group of Gaby is 'Chinese/Asian' and her education level is 'high school graduate'.
2. **Mention annotation** we mark the evidence for the profile properties as found in text. In the example given here, we would annotate 'finished high school' as an evidence for Gaby's education level and we would mark 'Chinese' to support the profile trait of Gaby being of Asian descent.

[9] We start with the documents from the 241 Gun Violence Corpus [60] incidents that were annotated with event mentions and we enrich them with additional 50 incidents whose participant shares a name with a participant in the original collection. This results in an initial pool of 291 incidents, which is later filtered as described below, leading to 260 incidents in total.

[10] https://docs.google.com/document/d/1rgTdrn-tPoJfPI25-5qOioznmj7un9pi-dO1FyV7pCk/edit#.

Table 4
Ambiguity statistics. Total number of unique instances in the partial data: 472, total instances in the full data: 5,408. *Meanambiguity* = *#uniqueInstances*/*#uniqueSFs*.

Partition	Modification	Unique SFs	Mean ambiguity
Partial	Original data	456	1.035
	Same first name 'John'	325	1.452
	Same last name 'Smith'	377	1.252
	Same name 'John Smith'	1	472
Full	Original data	5,329	1.015
	Same first name 'John'	3,547	1.525
	Same last name 'Smith'	3,557	1.520
	Same name 'John Smith'	1	5,408

A dedicated web-based tool was created to support this annotation.[11] Two linguistics Master students were hired as annotators for a day per week over a period of three months. The inter-annotator agreement for the structured annotation is 0.852, whereas the agreement on document- and sentence-level marking of evidence is 0.849 and 0.648, correspondingly. The remaining disagreement was resolved by using the annotation of the first annotator, as we observed that she was following the guidelines more consistently.

The additional annotation performed by our students was only performed on the documents and incidents that belong to the 'partial' dataset. For the full dataset, we only use the properties provided by the original data source.

Postprocessing In a postprocessing step, we remove: 1. documents and incidents that were disqualified by our annotators[12] 2. incidents without new annotation of structured data 3. documents without any annotation 4. participants with no name. In addition, we merged the incidents with different IDs which were identical, as well as the participants that appeared in multiple incidents.

Increase of ambiguity Besides including incidents with participants with the same name, within our experiments we systematically and artificially increase the ambiguity, simply by changing names of people in the structured data. We define four ambiguity levels:

1. Original data (no changes), containing 456 and 5,329 surface forms in the partial and the full dataset, respectively.
2. Same first name 'John' — the first name of all participants is changed to 'John', e.g., 'Paul McCartney' becomes 'John McCartney'. Note that after this change, the original names 'Mia McCartney' and 'Paul McCartney' both get changed to 'John McCartney', thus directly increasing the mean dataset ambiguity to nearly 1.5.
3. Same last name 'Smith' — the last name of all participants is changed to 'Smith'. For instance, 'Paul McCartney' becomes 'Paul Smith'. With this change, 'Paul McCartney' and 'Paul Lennon' both become 'Paul Smith', which increases the data ambiguity.
4. Same name 'John Smith' — all participants in the dataset share the same name, 'John Smith'.

Combining the four ambiguity levels and two data partitions leads to eight datasets in total. The ambiguity statistics for all eight datasets are presented in Table 4.

Table 5
Comparison of our attribute extractors against gold data for 1,000 GVDB articles.

	Precision	Recall	F1-score
Age	97.81	85.44	91.21
Ethnicity/race	40.00	8.70	14.29
Gender	81.78	57.14	67.28

5.3. Evaluation

We evaluate the accuracy of different methods for establishing identity of entities with the clustering metric Adjusted Rand Index (ARI). The ARI score between a system output and the gold clustering ranges between 0 and 1, where larger values stand for better clustering.

In addition, we perform intrinsic evaluation of the individual components in our end-to-end pipeline. Namely, we benchmark the property extractors by using the customary metrics of precision, recall, and F1-score. These evaluation results are provided in Section 5.4.

Regarding the profiling machines, we measure their intrinsic accuracy by evaluating them against 'correct' values in Wikidata. We also measure their correspondence to human expectations through a Jansen–Shannon divergence against the distribution of crowd judgments. We refer the reader to Section 7 for the results of these analyses.

5.4. Automatic attribute extraction

The local context of an entity consists of all property values that describe that entity in a document. We seek to understand the usefulness of profiling on top of this local context, generated with either perfect or imperfect attribute extraction. We have thus built the following automatic extraction strategies from text:

1. Proximity algorithm (Algorithm 1) which assigns spotted phrases in text to their closest mention of person, as long as this occurs in the same sentence. This strategy was applied to all properties, except for the gender.
2. Coreference algorithm (see Algorithm 2) looks for gender keywords in the coreferential phrases for a person.

Note that the property values are mapped to Wikidata Q-nodes to enable the use of profiling in a subsequent step.

We have benchmarked the automatic attribute extraction with both strategies against the gold extraction data we possess: see Table 5 for a comparison against known values in the Gun Violence Database (GVDB) [61] and Table 6 for a comparison against a subset of the SemEval-2018 task 5 data (see Section 5.2 for more details on this data).

As we discussed in Section 5.4, it is not trivial to compare the performance of our extractors directly to previous research on attribute extraction or slot filling. Fair comparison is prevented by differences in the test data (domain) and the considered properties. Given that the performance per property varies greatly and covers almost the entire spectrum of possible scores, it is particularly nonsensical to compare the average F1-scores over all properties, as this would directly be determined by the choice of properties.

Keeping in mind that the data differences persist, we make an attempt to compare individual overlapping properties in order to gain insight into the magnitude of our scores. For this purpose, let us consider them in relation to two previous slot filling systems: [26] and [27]. Although the majority of the properties we consider have no counterpart in these papers, we report

[11] Source code: https://github.com/cltl/AnnotatingEntityProfiles.

[12] Documents were disqualified if they were: too short, duplicates, or if they described a different incident. Incidents were disqualified if all their documents were disqualified.

Table 6
Benchmarking the attribute extractors on the gold data from the 456 documents of the SemEval dataset. For the attributes marked with '*', we do not have mention- or document-level annotation, hence the reported recall (and consequently F1-score) should be considered a lower bound for the 'real' score.

	Method	Precision	Recall	F1-score	#gold	#sys
Cause of death	Pattern	39.01	17.83	24.47	488	223
Past conviction	Pattern	9.52	25.00	13.79	32	84
Education level	Pattern	62.16	19.66	29.87	117	37
Ethnicity/race	Pattern	0.00	0.00	0.00	10	16
Religion	Pattern	50.00	11.76	19.05	17	4
Age group*	Pattern	93.20	25.98	40.63	739	206
Gender*	Coref	88.03	13.62	23.60	756	117
Gender*	Pattern	71.85	12.83	21.77	756	135
Birthplace	Pattern	0.00	0.00	0.00	6	1
Residence	Pattern	38.40	23.82	29.40	403	250

Algorithm 1: Attribute extraction: Proximity strategy.

```
Require: $attr, text, person\_names$
$regexes \leftarrow set\_regexes(attr)$
for all $person\_name \in person\_names$ do
    $spans[person\_name] \leftarrow findall(person\_name, text)$
    $coref\_spans[person\_name] \leftarrow coreference(spans, text)$
end for
for all $regex \in regexes$ do
    $matches \leftarrow findall(regex, text)$
    for all $match \in matches$ do
        $the\_person \leftarrow find\_closest\_person(match, spans, coref\_spans, sentence\_boundaries)$
        $the\_person[attr] += match$
    end for
end for
for all $person\_name \in person\_names$ do
    $person\_name[attr] \leftarrow find\_closest(person\_name[attr])$
end for
return $person\_names$
```

Algorithm 2: Attribute extraction: Coreference strategy.

```
Require: $attr, text, person\_names$
$regexes \leftarrow set\_regexes(attr)$
for all $person\_name \in person\_names$ do
    $spans[person\_name] \leftarrow findall(person\_name, text)$
    $coref\_spans[person\_name] \leftarrow coreference(spans, text)$
end for
for all $person\_name \in person\_names$ do
    $genders[person\_name] \leftarrow cooccurring(keywords, person\_name, spans, coref\_spans)$
end for
for all $person\_name \in person\_names$ do
    $gender[person\_name] \leftarrow most\_common(genders[person\_name])$
end for
return $gender$
```

comparisons of the properties that can be matched. The F1-score of extracting state(s) of residence by [26] and [27] is 12 and 31, respectively — whereas our performance on the SemEval-2018 task 5 dataset is 29.40; [26] and [27] extract age with F1-scores of 93 and 77 compared to our performance of 91.21 (GVDB) and 40.63 (SemEval)[13]; the cause of death F1-scores of these systems are 55 and 31, whereas ours is 24.47. This limited evidence tells us that our attribute extractors perform comparably to existing ones.

Algorithm 3: Probabilistic reasoner based on Jansen–Shannon entropy and Density-Based Spatial Clustering.

```
Require: $known\_attrs, profiles, person\_names, EPS$
$distances \leftarrow \emptyset$
for all $name1, name2 \in person\_names$ do
    $attrs[name1] \leftarrow known\_attrs(name1) \cup profiles(name1)$
    $attrs[name2] \leftarrow known\_attrs(name2) \cup profiles(name2)$
    $distances(name1, name2) \leftarrow avg\_js\_entropy(attrs[name1], attrs[name2])$
end for
$clusters \leftarrow DBSCAN\_clustering(distances, EPS)$
    return $clusters$
```

5.5. Profiler

In our extrinsic evaluation, we test the behavior of our AE method. The EMB method could not be applied to this task in its current state, because we have no ready embeddings for the long-tail entities found in text. An adapted version of this method that is able to compute an entity embeddings from text as a first step is needed in order to make this method applicable to NIL clustering too.

The AE profiler has been trained on Wikidata data about the 13 properties described in Section 5.2 (excluding the name attribute). To allow integration of the profiler with the information found in text, these properties and their corresponding values extracted from text were mapped manually to Wikidata URIs.

5.6. Reasoners

Once the attributes have been extracted, we use the following three strategies that compute clusters over the entities with extracted attributes:

1. Exact match (EX) — this reasoner clusters two entities only when all their attribute values are identical. Formally, $EX : (I_1 = I_2) \iff I_1(p) = I_2(p), \forall p \in P$.
2. No-clash (NC) — this reasoner establishes identity whenever two local representations have no conflicting properties, $NC : (I_1 = I_2) \iff \nexists p \in P, I_1(p) \neq I_2(p)$.
   We apply these two reasoners (EX and NC) on the properties extracted from text, but also on an extension of these properties provided by our profiler. In order to achieve the latter, we discretize the probabilities provided by the profiler based on a threshold parameter, τ. Namely, we keep the property values with a probability larger than τ, and discard the others.
3. Probabilistic reasoner (PR) — Since the profiler computes probabilistic distributions over values, we implemented a probabilistic reasoner that clusters entities based on the similarity of their attribute distributions. This reasoner first computes the average pairwise divergence (based on the Jansen–Shannon entropy) between the attribute distributions of all entities that share a name, resulting in a distance matrix.[14] Subsequently, a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [62] is run over this matrix to obtain clusters. The resulting clusters depend on a threshold called EPS, which determines the maximum distance allowed within a cluster. The PR reasoner is described in Algorithm 3.

[13] This F1-score is a lower bound given that the measured recall is lower than the real one, see Table 6. Considering the unreliability of this particular F1-score, it is more accurate to compare the gender attribute on the SemEval dataset against past work in terms of *precision*.

[14] We report results based on the mean divergence over all properties; using maximum instead of mean distance yielded comparable results.

Table 7
Clustering accuracy with various combinations of gold properties, using the 'exact' match clustering. Combinations of properties: p0=name baseline, p1=(name, educationlevel, causeofdeath), p2=(name,educationlevel, causeofdeath, residence), p3=(name, educationlevel, causeofdeath, residence, religion, ethnicity, pastconviction), p4=(name, educationlevel, causeofdeath, residence, religion, ethnicity, pastconviction, birthplace), p5=(name, age), p6=(name, age, gender), p7=(name, age, gender, death date), p8=(name, age, gender, death place), p9=(name, age, gender, death place, death date), p10=(name, causeofdeath, religion, ethnicity, pastconviction, age, gender, occupation, nativelanguage, politicalparty), all. Datasets: PD1=original partial data, PD2=partial data with same first name, PD3=partial data with same last name, PD4=partial data with all same names, FD1=original full data, FD2=full data with same first name, FD3=full data with same last name, FD4=full data with all same names. Some cells in the full datasets are empty due to unavailability of information.

	PARTIAL				FULL			
	PD1	PD2	PD3	PD4	FD1	FD2	FD3	FD4
p0	0.988	0.413	0.654	/	0.933	0.197	0.236	/
p1	0.846	0.598	0.767	0.006	/	/	/	/
p2	0.698	0.535	0.685	0.018	/	/	/	/
p3	0.679	0.522	0.666	0.018	/	/	/	/
p4	0.679	0.522	0.666	0.018	/	/	/	/
p5	0.993	0.701	0.818	0.005	0.951	0.354	0.358	0.012
p6	0.993	0.821	0.821	0.010	0.975	0.466	0.467	0.021
p7	0.997	0.904	0.932	0.037	0.976	0.472	0.473	0.037
p8	0.993	0.909	0.974	0.086	0.976	0.473	0.473	0.040
p9	0.997	0.916	0.983	0.093	0.976	0.473	0.473	0.040
p10	0.905	0.807	0.873	0.030	0.975	0.466	0.467	0.021
All	0.681	0.636	0.681	0.384	/	/	/	/

Table 8
Clustering accuracy with various combinations of gold properties, using the 'no-clash' match clustering. Combinations of properties: p0=name baseline, p1=(name, educationlevel, causeofdeath), p2=(name,educationlevel, causeofdeath, residence), p3=(name, educationlevel, causeofdeath, residence, religion, ethnicity, pastconviction), p4=(name, educationlevel, causeofdeath, residence, religion, ethnicity, pastconviction, birthplace), p5=(name, age), p6=(name, age, gender), p7=(name, age, gender, death date), p8=(name, age, gender, death place), p9=(name, age, gender, death place, death date), p10=(name, causeofdeath, religion, ethnicity, pastconviction, age, gender, occupation, nativelanguage, politicalparty), all. Datasets: PD1=original partial data, PD2=partial data with same first name, PD3=partial data with same last name, PD4=partial data with all same names, FD1=original full data, FD2=full data with same first name, FD3=full data with same last name, FD4=full data with all same names. Some cells in the full datasets are empty due to unavailability of information.

	PARTIAL				FULL			
	PD1	PD2	PD3	PD4	FD1	FD2	FD3	FD4
p0	0.988	0.413	0.654	/	0.933	0.197	0.236	/
p1	0.987	0.611	0.783	0.002	/	/	/	/
p2	0.976	0.634	0.861	0.012	/	/	/	/
p3	0.976	0.632	0.865	0.013	/	/	/	/
p4	0.976	0.632	0.865	0.013	/	/	/	/
p5	0.991	0.687	0.802	0.005	0.936	0.348	0.352	0.012
p6	0.991	0.807	0.804	0.010	0.965	0.459	0.459	0.019
p7	0.995	0.852	0.861	0.034	0.965	0.462	0.462	0.030
p8	0.991	0.863	0.903	0.078	0.965	0.463	0.463	0.034
p9	0.995	0.869	0.913	0.088	0.965	0.463	0.463	0.035
p10	0.991	0.840	0.886	0.025	0.965	0.459	0.459	0.019
All	0.980	0.895	0.932	0.366	/	/	/	/

6. Extrinsic, end-to-end, evaluation

In this section we present the results of our experiments on using the profiler within an end-to-end pipeline to tackle the task of NIL clustering, thus addressing the hypotheses C1 through C6. Namely, Section 6.1 shows the performance of our pipeline with perfect and imperfect attribute extraction, thus addressing C1 and C2. In Section 6.2, we provide the results of running the profiler on top of these attribute extractors (C3 and C4). Finally, we analyze the impact of the task ambiguity on the clustering performance (C5) and its relation to the effectiveness of our profiler (C6).

6.1. Using explicit information to establish identity

We hypothesized that the performance of clustering by attribute reasoning depends on two factors (C1): availability of information and quality of extraction. An ideal availability of information and perfect extraction would lead to a perfect accuracy on the task of establishing identity of entities.

We first present the clustering performance of various combinations of perfectly extracted attributes in Tables 7 and 8. Namely, besides the name baseline (p0) and the union of all properties, we consider additional ten combinations of properties. We note that the sets p1 through p4 rely on properties annotated in documents by our students; for these attributes, we know whether they are covered in the annotated documents. The combinations p5 through p9 rely on the structured incident data as found on the GVA website. For these sets of properties, we make an assumption that they are consistently mentioned in all reporting documents. As we do not know how often this assumption holds, the obtained scores for p5–p9 should thus be seen as upper bound results for the real scores. Finally, we report performance of p10, which represents the set of properties that were successfully mapped to background knowledge, and will be used for a comparison to the profiler later in this paper.

This analysis provides an insight into the availability and the diversity of information about an entity across documents, as well as into the discriminative power of this information. We observe that correct extraction of the properties leads to almost perfect accuracy on the original data, but notably lower accuracy on the more ambiguous subsets. Hence, assuming perfect attribute extraction, the properties found in text would be mostly sufficient to establish the identity of entities in the original dataset, but this information becomes increasingly insufficient as the ambiguity grows.[15] The results also show that certain attributes, such as age and gender, have very large discriminative power, as long as they are consistently reported across documents.

Comparing the two tables reveals crucial differences in the behavior of our reasoners. We see that including more attributes is not necessarily beneficial for the exact reasoner (Table 7, attribute sets p1–p4). For example, including the person's state of residence (p2) causes a decline in performance on PD1, PD2, and PD3. This decline is to be expected given that the exact reasoner regards two local representations to be identical only when all their property values are identical. Since properties like residence are not consistently mentioned across documents, the exact reasoner switches its judgment to non-identity when one's residence is mentioned in one document and not in another. However, adding properties is beneficial for the no-clash reasoner, because it is more robust with respect to missing information and decides that two representations are identical as long as no property value is contradictory between them.

Next, in Table 10, we show the clustering accuracy when automatic attribute extraction is employed. While the no-clash reasoner comes closer to the perfect extraction performance than the exact reasoner, the clustering performance of the automatic attribute extractors is consistently lower than that of the perfect attribute extractors, as expected in our hypothesis C2. This difference in clustering performance grows together with the ambiguity of data. These findings are not surprising given the

[15] We unfortunately do not know if some of the attributes (e.g., gender and age) presented in the gold data occur in each document, hence the scores presented here might be higher than the real ones.

Table 9
Inspection of the effect of profiling on the clustering performance, on top of gold attribute extraction. Datasets: PD1=original partial data, PD2=partial data with same first name, PD3=partial data with same last name, PD4=partial data with all same names, FD1=original full data, FD2=full data with same first name, FD3=full data with same last name, FD4=full data with all same names. The set of properties used by the baselines and by the profiler corresponds to p10 in Table 7 and Table 8. We report results of applying the profiler in combination with each of the baselines, by first discretizing its probability distribution with a threshold, τ. In addition, we report results of using the probability distributions as provided by the profiler, in combination with Jansen–Shannon entropy and DBSCAN clustering, which corresponds to the probabilistic reasoner (PR) described in section 5.6. We vary the clustering coefficient, EPS, between 0.01 and 0.5.

	PARTIAL				FULL		
	PD1	PD2	PD3	PD4	FD1	FD2	FD3
Name baseline (p0)	0.988	0.413	0.654	/	0.933	0.197	0.236
Exact (EX) reasoner	0.905	0.807	0.873	**0.030**	**0.975**	**0.466**	**0.467**
+ profiler ($\tau = 0.99$)	0.905	0.807	0.873	**0.030**	0.973	0.384	0.378
+ profiler ($\tau = 0.90$)	0.911	0.810	0.870	0.029	0.973	0.384	0.378
+ profiler ($\tau = 0.75$)	0.911	0.810	0.870	0.029	0.949	0.313	0.309
+ profiler ($\tau = 0.51$)	0.910	0.810	0.869	0.029	0.965	0.367	0.360
No-clash (NC) reasoner	**0.991**	0.840	**0.886**	0.025	0.965	0.459	0.459
+ profiler ($\tau = 0.99$)	0.989	0.839	0.885	0.022	0.963	0.369	0.358
+ profiler ($\tau = 0.90$)	0.977	**0.843**	0.857	0.019	0.963	0.371	0.360
+ profiler ($\tau = 0.75$)	0.960	0.836	0.848	0.023	0.934	0.298	0.294
+ profiler ($\tau = 0.51$)	0.925	0.811	0.857	0.020	0.934	0.354	0.347
PR reasoner, EPS = 0.01	0.906	0.808	0.871	**0.030**	0.973	0.384	0.378
PR reasoner, EPS = 0.05	0.914	0.808	0.871	0.026	0.934	0.290	0.285
PR reasoner, EPS = 0.1	0.950	0.811	0.839	0.002	0.934	0.273	0.276
PR reasoner, EPS = 0.2	0.987	0.476	0.673	0.000	0.933	0.198	0.236
PR reasoner, EPS = 0.5	0.988	0.413	0.654	0.000	0.933	0.197	0.236

Table 10
Inspection of the effect of profiling on the clustering performance, on top of automatic attribute extraction. Datasets: PD1=original partial data, PD2=partial data with same first name, PD3=partial data with same last name, PD4=partial data with all same names, FD1=original full data, FD2=full data with same first name, FD3=full data with same last name, FD4=full data with all same names. The set of properties used by the baselines and by the profiler corresponds to p10 in Table 7 and Table 8. We report results of applying the profiler in combination with each of the baselines, by first discretizing its probability distribution with a threshold, τ. In addition, we report results of using the probability distributions as provided by the profiler, in combination with Jansen–Shannon entropy and DBSCAN clustering, which corresponds to the probabilistic reasoner (PR) described in section 5.6. We vary the clustering coefficient, EPS, between 0.01 and 0.5.

	PARTIAL				FULL		
	PD1	PD2	PD3	PD4	FD1	FD2	FD3
Name baseline (p0)	**0.988**	0.413	0.654	/	**0.933**	0.197	0.236
Exact (EX) reasoner	0.609	0.385	0.534	**0.002**	0.780	0.212	0.237
+ profiler ($\tau = 0.99$)	0.622	0.390	0.545	0.001	0.784	0.211	0.237
+ profiler ($\tau = 0.90$)	0.637	0.398	0.555	0.001	0.787	0.211	0.238
+ profiler ($\tau = 0.75$)	0.666	0.397	0.576	0.001	0.787	0.206	0.231
+ profiler ($\tau = 0.51$)	0.693	0.401	0.576	0.001	0.848	0.215	0.250
No-clash (NC) reasoner	0.944	**0.506**	**0.768**	**0.002**	0.910	0.229	0.258
+ profiler ($\tau = 0.99$)	0.943	0.502	0.745	0.001	0.916	**0.235**	**0.274**
+ profiler ($\tau = 0.90$)	0.932	0.485	0.713	0.001	0.920	0.229	0.271
+ profiler ($\tau = 0.75$)	0.898	0.459	0.657	0.001	0.910	0.225	0.260
+ profiler ($\tau = 0.51$)	0.825	0.410	0.620	0.000	0.897	0.210	0.245
PR reasoner, EPS = 0.01	0.645	0.398	0.555	0.001	0.786	0.211	0.238
PR reasoner, EPS = 0.05	0.726	0.396	0.595	0.001	0.849	0.210	0.244
PR reasoner, EPS = 0.1	0.858	0.384	0.631	0.000	0.901	0.201	0.236
PR reasoner, EPS = 0.2	0.974	0.413	0.654	0.000	0931	0.197	0.236
PR reasoner, EPS = 0.5	**0.988**	0.413	0.654	0.000	**0.933**	0.197	0.236

benchmark results of these automatic extractors presented in Section 5.4.

In addition, we observe that both for gold and for automatically extracted information, the improvement in terms of clustering compared to the name baseline is larger when the ambiguity of data is larger.

6.2. Profiling implicit information

The profiler can be seen as a 'soft' middle ground between the exact reasoner and the no-clash reasoner. The former establishes identity only when all known attributes between two local representations are identical, and the latter — whenever two local representations have no conflicting properties. The profiler relies on background knowledge in order to fill the gaps for the unknown properties instead of making a hard decision. We hence expect the profiler to be superior over the above two baselines, as it employs external knowledge to bring closer or further two ambiguous representations. We hypothesized that the role of the profiler is much more important when the attribute extraction is imperfect (hypotheses C3 and C4).

Tables 9 and 10 show the impact of the profiler when applied on top of either perfect or imperfect attribute extraction.[16] We note that the results reported here use the following set of attributes: name, religion, ethnic group, cause of death, gender, occupation, age group, native language, and political party. These properties correspond to the property set *p10* in Tables 7 and

[16] We leave out the results on the full dataset with maximum ambiguity as these are consistently very low.

Table 11
Number of clusters for different values of the EPS parameter. Datasets: PD1=original partial data, PD2=partial data with same first name, PD3=partial data with same last name, PD4=partial data with all same names, FD1=original full data, FD2=full data with same first name, FD3=full data with same last name, FD4=full data with all same names.

EPS		PD1	PD2	PD3	PD4	FD1	FD2	FD3
Gold	0.01	503	475	489	46	5,344	4,354	4,113
	0.05	498	467	480	30	5,335	4,238	3,932
	0.1	480	439	446	4	5,335	4,220	3,926
	0.2	459	360	390	1	5,329	3,610	3,563
	0.5	456	325	377	1	5,328	3,547	3,557
Auto	0.01	608	526	566	31	7,597	5,865	5,954
	0.05	578	475	525	19	6,768	5,027	5,091
	0.1	521	399	451	5	5,967	4,214	4,280
	0.2	464	332	384	1	5,376	3,574	3,600
	0.5	456	325	377	1	5,328	3,547	3,557

8. Notably, as the properties: native language, occupation, and political party do not occur in text, their values are always filled implicitly by the profiler. The expectations for the other properties are filled by the profiler only when they are not extracted from text.

As discussed previously, in order to experiment with the profiler in combination with the EX and NC reasoners, the probabilities produced by the profiler are first discretized by using a cut-off threshold, τ. Larger values of this threshold mean stricter criteria for inclusion of a certain attribute value; hence, a τ value of 1.0 excludes all output from the profiler, whereas lower τ values allow for more profiler expectations to be included. We also test the usefulness of the profiler in combination with the probabilistic reasoner, PR; in this case, we use the probability distribution as produced by the profiler, in combination with the extracted property values from text. Here, to acknowledge the effect of the clustering parameter of maximum distance (EPS) in the DBSCAN algorithm, we report results with five different EPS values, ranging between 0.05 and 0.5 (higher EPS values lead to less clusters with larger sizes). The number of clusters for different values of the EPS parameter are given in Table 11.

Table 9 shows that enhancing the gold properties by profiling yields comparable results to the ones obtained by the baselines, which corresponds to our hypothesis C3. This observation holds for all three reasoners: exact, no-clash, and probabilistic reasoner. Given that the results of clustering by reasoning over the perfectly extracted properties are relatively high (Tables 7 and 8), there is little room to improve them by profiling. Still, enhancing these gold properties by profiling yields slight improvement over the results of the extractors in several occasions. For instance, combining the no-clash reasoner with profiling at $\tau = 0.90$ (i.e., keeping the profiling values with a probability of 0.90 or higher), leads to the best score on the PD2 dataset. Similarly, the profiler increases the performance on top of the exact baseline for the datasets PD1 and PD2. While for the other datasets the profiler does not improve the baseline performance, we note that its output is fairly robust — even when much of the profiler output is included (e.g., with $\tau = 0.51$), the results do not decline to a large extent.

We expect that the profiler has a larger effect when combined with automatic attribute extraction (C4). While we do not observe significant improvement over the baselines, the results in Table 10 demonstrate that the profiler has certain impact when applied in combination with imperfect attribute extraction. Especially, we observe that the profiler consistently improves the performance on top of the exact reasoner. Furthermore, this improvement becomes larger when more of the profiling values are included, i.e., the improvement on the exact reasoner is inversely proportional with the probability threshold τ. This observation means that the profiler is able to fulfill its role of normalizing the attribute values found in different documents. Namely, as these documents are written independently from each other, certain attribute values are reported only in a subset of them. In addition, the low recall of our automatic extraction tools might increase this inconsistency of information between documents. The results show that the profiler is able to make this reasoner more robust and normalize certain knowledge gaps.

Profiling in combination with the no-clash reasoner yields comparable results to those of only using the automatic extraction. Concretely, profiling improves the performance of the no-clash reasoner for all full datasets (FD1, FD2, and FD3), whereas it decreases the baseline performance on all partial datasets. In general, the profiling results seem to be fairly robust when combined with this reasoner as well.

The probabilistic reasoner also yields certain promising results on the full datasets, whereas its best scores on the partial datasets are obtained for the highest value of the EPS distance parameter (0.5), and correspond to the results of the name baseline.

What limits the performance of our profiler on the NC reasoner? In Section 5.4, we observed that the performance of our extractors, especially in terms of recall, is relatively low. Consequently, the extracted local contexts from text are largely incomplete, and in some cases contain incorrect values. As the output of the profiler is dependent on the input it receives, this imperfection of the extracted information would directly influence the usefulness of the generated profiles. Namely, a very sparse/incomplete input (caused by low extraction recall) could lead to a profile that represents a more generic group, whereas incorrect input (caused by low precision) might generate a profile that represents an entirely different group of entities. Future work should investigate whether an extended set of attributes and a different kind of attribute extractor yield similar results for the hypotheses C3 and C4.

6.3. *Analysis of ambiguity*

We expect that the clustering performance is counter-proportional to the ambiguity of a dataset (C5), i.e., higher ambiguity leads to lower clustering performance. This is a clear trend that is visible in all result tables. The clustering on the original dataset with minimal ambiguity is close to perfect, whereas the performance of clustering for the datasets with maximum ambiguity is close to zero.

We also hypothesized that the impact of the profiler is higher when there is more ambiguity (C6). Tables 9 and 10 show no clear relation between the usefulness of the profiler and the data ambiguity. Future work should investigate whether this conclusion is confirmed for a larger set of properties.

7. Intrinsic analysis of the profilers

In this section, we investigate the intrinsic behavior of the profilers, by comparing them against factual data from Wikidata, as well as against human judgments collected with a crowdsourcing task. We also investigate the relation of the profiling performance to various properties of the data, such as its entropy and value size. These investigations provide evidence for the hypotheses P1–P5.

7.1. *Comparison against factual data*

7.1.1. *Data*

No existing dataset is directly suitable to evaluate profiling. We therefore chose People, since data is plentiful, people are multifaceted, and it is easy to spot problematic generalizations.

Table 12
Numbers of examples (n_{ex}), categories (v_i), and entropy (H_i and $H_{i'}$) per facet of People in our training data. We limit v_i to 3,000 to restrict the complexity of the value space, but also to mimic the simplification aspect of cognitive profiling.

Attribute	*PERSON*				*POLITICIAN*				*ACTOR*			
	n_{ex}	v_i	H_i	$H_{i'}$	n_{ex}	v_i	H_i	$H_{i'}$	n_{ex}	v_i	H_i	$H_{i'}$
Educated at	273,096	3,000	9.28	0.80	22,461	3,000	9.73	0.84	5,047	883	7.56	0.77
Sex or gender	2,403,980	11	0.64	0.18	168,758	5	0.50	0.25	75,980	5	1.00	0.50
Citizenship	1,546,757	995	5.28	0.53	152,131	335	5.07	0.61	57,570	187	5.12	0.68
Native language	41,760	141	1.70	0.24	16,818	33	1.08	0.21	4,273	29	0.41	0.08
Position held	177,302	3,000	7.44	0.64	101,766	1,701	7.08	0.66	244	25	0.96	0.21
Award received	154,275	3,000	7.97	0.69	10,588	546	6.82	0.75	2,880	297	6.60	0.80
Religion	32,311	341	3.24	0.38	2,414	127	3.99	0.58	164	24	2.47	0.56
Political party	158,105	3,000	7.28	0.63	82,617	2,456	7.26	0.64	232	53	3.23	0.58
Work location	68,602	1,989	6.25	0.57	30,320	272	5.07	0.63	116	41	3.99	0.74
Place of death	350,720	3,000	7.93	0.68	29,071	3,000	8.39	0.73	9,377	2,169	8.33	0.75
Place of birth	927,089	3,000	7.64	0.66	59,627	3,000	7.27	0.63	39,694	3,000	8.55	0.74
Cause of death	21,926	499	5.35	0.60	1,408	115	4.75	0.69	1,039	82	4.22	0.66
Lifespan range	922,634	55	1.89	0.33	79,346	39	1.68	0.32	19,055	11	1.77	0.49
Century of birth	1,975,197	43	1.36	0.25	140,087	22	1.48	0.33	61,506	11	0.56	0.16

We defined three typed datasets: people, politicians, and actors, each with the same stereotypical facets, such as nationality, religion, and political party, that largely correspond to some facets central in social psychology research. We created data tables by extracting facets of people from Wikidata. Table 12 lists all attributes, each with its number of distinct categories v_i, total non-empty values (n_{ex}), and entropy values (H_i and $H_{i'}$) on the training data.

The goal is to dynamically generate expectations for the same set of 14 facets in each dataset. We evaluate on multiple datasets to test the sensitivity of our models to the number of examples and categories. The largest dataset describes 3.2 million people, followed by the data on politicians and actors, smaller by an order of magnitude. As pre-trained embeddings are only available for a subset of all people in Wikidata (see Section 4.4.4), we cannot evaluate EMB directly on these sets. Hence, to facilitate a fair comparison of both our models on the same data, we also define smaller data portions for which we have pre-trained embeddings. We randomly split each of the datasets into training, development, and test sets at 80-10-10 ratio.

7.1.2. Quantification of the data space

We quantify aspects of profiling through the set of possible outcomes and its relation to the distribution of values.

The total size of the data value space is $d_{size} = \prod_{i=1}^{n} v_i$, where n is the number of attributes and v_i is the size of the category vocabulary for an attribute x_i (e.g. $v_i = |\{\text{Swiss, Dutch}\dots\}|$ for $x_i = nationality$). We define the **average training density** as the ratio of the total data value size to the overall number of training examples n_{ex}: $d_{avg-d} = d_{size}/n_{ex}$. As an illustration, we note that the full dataset on People has $d_{size} = 10^{39}$ and $d_{avg-d} = 10^{32}$.

For the ith attribute x_i, the entropy H_i of its values is their 'dispersion' across its v_i different categories. The entropy for each category j of x_i is computed as $-p_{i,j}logp_{i,j}$, where $p_{i,j} = n_{ex}(i,j)/n_{ex}(i)$. The **entropy** of x_i is then a sum of the individual category entropies: $H_i = -\sum_{j=1}^{v_i} p_{i,j}logp_{i,j}$, whereas its **normalized entropy** is limited to [0, 1]: $H_{i'} = H_i/log_2(n_{ex}(i))$. Entropy is a measure of informativeness: when $H_{i'} = 0$ there is only one value for x_i; when all values are equally spread the entropy is maximal, $H_{i'} = 1$ (with no MFV).

Of course, we do not know the true distribution but only that of the sparse input data. Here we assume our sample is unbiased. Table 12 shows that, e.g., *educated at* consistently has less instance values and a 'flatter' value distribution (= higher $H_{i'}$) than *sex or gender*, where the category *male* is dominant on any dataset, except for actors. The entropy and the categories size together can be seen as an indicator for the relevance of a facet for a dataset, e.g., $H_{i'}$ and v_i of *position held* are notably the lowest for actors. We expect MFV to already perform well on facets with low entropy, whereas higher entropy to require more complex dependencies.

7.1.3. Results

We evaluate by measuring the correctness of predicted (i.e., top-scoring) attribute values against their (not provided) true values, evaluated only on exemplars that were not included in the training data.

Table 13 provides the results of our methods and baselines on the three smaller datasets that contain embeddings (the full datasets yielded similar results for MFV, NB, and AE). We observe that AE and EMB outperform the baselines on almost all cases, as hypothesized in P1. As expected (hypothesis P3), we see lower (or no) improvement over the baselines for cases with low entropy (e.g., *sex or gender* and *lifespan range*) compared to attributes with high entropy (e.g., *award received*). We also note that the accuracy of profiling per facet correlates inversely with its vocabulary size v_i (hypothesis P4).

The superiority of the neural models over the baselines means that capturing complex inter-facet dependencies improves the profiling ability of machines. Moreover, while the two neural methods perform comparably on average, there are differences between their accuracy on individual facets (e.g., compare *award received* and *native language* on any dataset). To gain further insight, we analyze the predictions of our models on an arbitrarily chosen instance from our dataset, namely, the Ohio politician Brad Wenstrup.[17] Brad is a male American citizen, member of the republican party, born in Cincinnati, Ohio in the 20th century, educated at the University of Cincinnati, and has held the position of a United States representative. Both neural systems correctly predict (as a first choice) Brad's country of citizenship, position held, work location, and century. The AE system is able to predict the political party and his education better than the EMB one; whereas the EMB model is superior over AE when it comes to Brad's gender. In addition, EMB ranks the correct place of birth third in terms of probability, while AE's top 3 predictions for this attribute are incorrect, i.e., these are places in the neighboring state of Michigan. These differences might be due to the main architectural difference between these two methods: EMB's input embedding contains much more information (both signal and noise) than what is captured by the 14 facets in the AE.

How does a profile improve (or not) with increasing input? To investigate our hypothesis P5, we analyze both top-1 and top-3 accuracies of AE for predicting a facet value against the number of other known facets provided at test time. Fig. 2 shows

[17] https://www.wikidata.org/wiki/Q892413.

Table 13
Top-1 accuracies for the both neural methods and the two baselines on the smaller datasets. For each dataset-facet pair, we emphasize the best result. Our neural methods, especially EMB, outperform the baselines. Entropy and vocabulary sizes can partially explain deltas in accuracies on individual facets.

Attribute	*PERSON*				*POLITICIAN*				*ACTOR*			
	MFV	NB	AE	EMB	MFV	NB	AE	EMB	MFV	NB	AE	EMB
Educated at	4.41	9.22	13.20	**22.45**	2.57	6.88	**13.14**	9.47	11.32	15.09	3.77	**46.43**
Sex or gender	82.61	81.76	82.37	**95.83**	85.15	84.10	83.23	**94.79**	49.71	57.97	55.20	**89.06**
Citizenship	29.10	57.36	66.49	**78.49**	18.27	46.75	72.94	**77.96**	17.99	39.94	60.77	**65.05**
Native language	44.70	69.44	**87.63**	33.33	46.67	88.89	**93.33**	83.33	**95.00**	**95.00**	**95.00**	91.67
Position held	8.44	32.92	**45.66**	21.43	15.47	28.93	**45.03**	41.18	50.00	50.00	50.00	**100.0**
Award received	4.98	15.95	21.56	**37.50**	3.85	10.58	18.27	**26.09**	14.29	14.29	23.81	**42.86**
Religion	27.52	40.83	45.48	**71.43**	27.08	42.71	52.08	**56.52**	40.00	40.00	60.00	**66.67**
Political party	13.18	29.67	42.08	**47.06**	9.41	22.78	34.28	**37.59**	**50.00**	**50.00**	**50.00**	0.0
Work location	22.47	57.18	**64.49**	60.00	22.22	69.90	**83.09**	75.00	0.00	0.00	0.00	0.00
Place of death	4.09	25.09	28.20	**36.84**	2.81	8.03	17.27	**25.81**	9.78	17.58	18.48	**33.93**
Place of birth	2.85	33.01	32.07	**49.04**	1.88	**54.62**	23.59	52.21	5.31	11.28	16.87	**36.21**
Cause of death	23.80	24.13	**24.24**	15.38	32.76	37.93	24.14	**71.43**	33.33	33.33	20.00	**45.00**
Lifespan range	41.76	**43.56**	41.69	42.03	41.30	40.68	38.51	**48.75**	36.73	39.17	**45.92**	43.33
Century of birth	82.04	85.45	84.94	**89.53**	76.13	80.13	83.14	**85.79**	**93.62**	93.60	89.56	92.67

Table 14
Human evaluation results per attribute: number of values (v_i), entropy (H_i), normalized entropy ($H_i\prime$), mean judgments entropy (J_i), divergences of: MFV, NB, and AE.

Attribute	v_i	H_i	$H_i\prime$	J_i	MFV	NB	AE
Century of birth	5	0.40	0.92	10^{-8}	0.13	**0.12**	**0.12**
Religion	4	0.63	1.26	10^{-10}	**0.05**	0.09	0.06
Sex or gender	2	0.70	0.70	10^{-14}	0.04	**0.02**	**0.02**
Place of death	8	0.80	2.40	0.05	0.51	0.20	**0.16**
Lifespan range	10	0.81	2.68	0.02	0.29	**0.09**	**0.09**
Place of birth	8	0.83	2.48	0.01	0.39	0.26	**0.24**
Work location	10	0.84	2.80	0.03	0.49	**0.28**	0.30
Occupation	9	0.92	2.90	0.06	0.37	0.36	**0.32**
Educated at	9	0.92	2.91	0.06	0.39	0.25	**0.23**
Political party	2	1.00	1.00	0.02	0.17	**0.06**	**0.06**

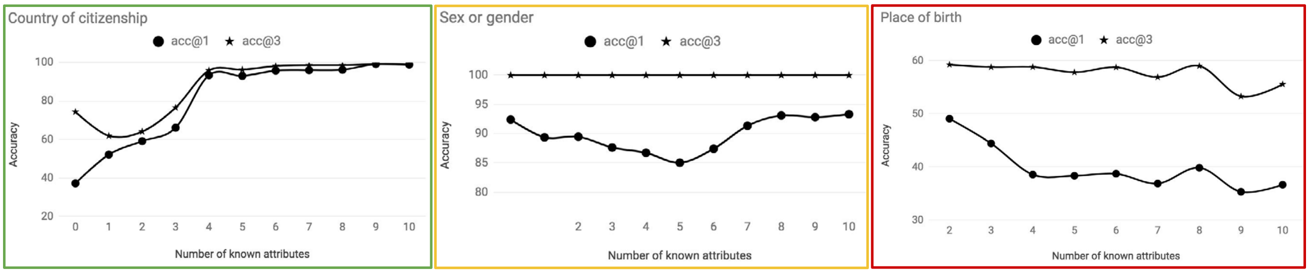

Fig. 2. Dependency of the accuracy of profiling politicians on the number of known facets: a positive correlation for *country of citizenship* (left Figure), no correlation for *sex or gender* (center), and a slightly negative one for *place of birth* (right).

examples of all three possible correlations (positive, negative, and none) for politicians. These findings are in line with conclusions from social psychology (cf. *Profiles*): knowing more facets of an instance might trigger a shift of the original profile, and it might be correct or incorrect, as defined in [51]. Generally, we expect that attributes with large v_i, like *place of birth*, will suffer as input exemplars become more specified and granularity becomes tighter, while facets with small v_i would benefit from additional input. Fig. 2 follows that reasoning, except for *sex or gender*, whose behavior is additionally influenced by low entropy (0.25) and strong frequency bias to the *male* class. Further research should study the correlation between different attributes, and seek ways to describe their dependencies in relation to properties of the underlying instance data, like entropy or value size.

7.2. Comparison against human expectations

Given that in most AI applications information is created for humans, a profiler has to be able to mimic human expectations. We thus compare our neural profiles to profiles generated by crowd workers.

7.2.1. Data

We evaluate on 10 well-understood facets describing American citizens. For each facet, we generated a list of 10 most frequent values among American citizens in Wikidata, and post-processed them to improve their comprehensibility. We collected 15 judgments for 305 incomplete profiles with the Figure Eight crowdsourcing platform. The workers were instructed to choose 'None of the above' when no choice seemed appropriate, and 'I cannot decide' when all values seemed equally intuitive. We picked reliable, US-based workers, and ensured US minimum wage ($7.25) payment.

Given that there is no 'correct' answer for a profile and our annotators' guesses are influenced by their subjective experiences, it is reasonable that they have a different intuition in some cases. Hence, the relatively low mean Krippendorff (1980) alpha agreement per property (0.203) is not entirely surprising. We note that the agreement on the high-entropy attributes is typically lower, but tends to increase as more facets were provided. Overall, the annotators chose a concrete value rather than being indecisive ('I cannot decide') for the low-entropy more often than the high-entropy facets. When more properties were provided, the frequency of indecisiveness on the high-entropy facets declined.

7.2.2. Results

When evaluating, 'None of the above' was equalized to any value outside of the most frequent 10, and 'I cannot decide' to a $(1/N)$-th vote for each of the values. The human judgments per profile were combined in a single distribution, and compared to the system distribution by using Jansen–Shannon divergence (*JS-divergence*).[18] We evaluate the profiles generated by our AE and the baselines; EMB could not be tested on this data since most inputs do not have a corresponding Wikipedia page and pre-trained embeddings.

The divergence between our AE system and the human judgments was mostly lower than that of the baselines (Table 14), which supports our hypothesis P2. The divergences for any system have a strong correlation with (normalized) entropy, confirming our previous observation that high-entropy attributes pose a greater challenge (hypothesis P3). We also computed precision, recall, and F1-score between the classes suggested by our system and by the annotators, and observed that it correlates inversely with the entropy in the data (H_i), as well as the entropy of the human judgments (J_i).

The results show that our AE can capture human-like expectations better than the two baselines, and that mimicking human profiling is more difficult when the entropy is higher. While parameter tuning and algorithmic inventions might improve the profiling accuracy further, it is improbable that profiles learned on factual data would ever equal human performance. Some human expectations are culturally projected and do not correspond to episodic facts. Future work should seek novel solutions for this challenge.

8. Discussion and limitations

8.1. Summary of the results

Identity clustering is nearly ideal assuming perfect attribute extraction, indicating that the information available in text often suffices to establish identity, as long as the ambiguity is low (hypothesis C1). As expected, the clustering performance declines when the extraction is imperfect (C2).

Given that the clustering based on the gold properties is relatively high, there is little room for improvement by profiling in this case. It is thus no surprise that the profiler has no visible effect when combined with gold properties (C3). Profiling is able to fill certain knowledge gaps when combined with automatic extraction of properties (C4). Concretely, profiling consistently has a positive impact when using exact reasoning over property values. Profiling is also beneficial when applied together with the no-clash reasoner on the full datasets (FD1–FD3), whereas it performs slightly worse than the no-clash reasoner on the partial datasets (PD1–PD4). Overall, the profiler is fairly robust to different hyperparameter values and degrees of data ambiguity.

The low performance of our automatic attribute extractors, especially in terms of recall, affects our results, considering that the effectiveness of our profiler is directly conditioned on the completeness and accuracy of its input. Incorrect or incomplete inputs lead to more generic or even wrong profiles, and the decisions on identity based on these profiles are likely to be consequently wrong as well.

The datasets with larger ambiguity pose a greater challenge for all approaches (C5). However, we did not see a clear relation between the usefulness of our profiler and the data ambiguity (C6). A possible explanation for this finding lies in the low number of properties considered, as well as in their usefulness to discriminate long-tail entities in the test domain. Future work should investigate whether these findings generalize for an extended set of properties.

The intrinsic evaluation of our profiling methods demonstrated their ability to largely mimic the instantial data in Wikidata (P1), as well as human judgments (P2). Notably, their performance per attribute fluctuates dramatically, but this variance can be easily predicted by the factors of entropy, value size, and other known attributes (hypotheses P3, P4, and P5). It remains a future task to relate this accuracy variance to the clustering performance of the profiler.

8.2. Bridging knowledge between text and KBs

A large challenge during these experiments lay in harmonizing the knowledge found in the text documents with the one found in the chosen knowledge base, Wikidata:

(1) Discrepancy of properties There is little overlap between the attributes that are found in background knowledge and those found in text. For example, descriptions of the kind of area in which an entity lives (e.g., is a city part dangerous or safe), are prominent in text and useful for identity reasoning; unfortunately, this kind of information is not present in Wikidata. On the other hand, one's height or political affiliation is found in Wikidata, but seldom mentioned in text.

(2) Discrepancy of property values To illustrate this gap, consider that most causes of death in Wikidata are natural and do not correspond to those in the gun violence domain, where most people died of a gunshot wound. In addition, the number of values for certain properties, such as birthplace, in Wikidata is quite large: it is non-trivial to map these values to those extracted from text.[19]

(3) Discrepancy of world expectations Expectations learned from a knowledge base do not always fit those within a local crime domain. As an example, people in Wikidata are typically highly educated, which is probably not representative for the typical *education level* of the participants in the gun violence incidents. Similarly for *occupations*: typical professions in Wikidata, such as politicians or actors, are unlikely to be found in gun violence incident descriptions.

These discrepancies are largely due to the different world captured by the two proxies: our documents describe local crimes, whereas Wikidata stores global events and well-known people. Learning models about the 'head' entities in Wikidata and applying these to the 'long-tail' entities in the gun violence domain required manual engineering and has been achieved at the expense of information loss, which decreased the discriminating power of our profiling machines. Future work should consider learning expectations that resemble the target domain closer, e.g., using gun violence domain data to both train the profiles and apply them.

[18] We considered the following metrics: JS-divergence, JS-distance, KL-divergence, KL-divergence-avg, KL-divergence-max, and cosine distance [64]. The agreement was very high, the lowest Spearman correlation being 0.894.

[19] In our experiments, we opted for several (typically between two and ten) attribute values on a coarser granularity level. Mapping these to Wikidata values required a substantial manual effort, as the connection in the structured knowledge was not consistently stored.

8.3. Limitations of profiling by NNs

Our experiments show the natural power of neural networks to generalize over knowledge and generate profiles from data independent of schema availability. Techniques like dropout and oversampling further boost their ability to deal with missing or underrepresented values. Ideally these profiling machines can be included in an online active representation system to create profiles on the fly, while their modularity allows easy retraining in the background when needed.

Still, it is essential to look critically beyond the accuracy numbers, and identify the strengths and weaknesses of the proposed profiling methods. Limitations include: 1. **continuous values**, such as numbers (e.g., age) or dates (e.g., birth date), need to be categorized before being used in an AE[20]; 2. AE cannot natively handle **multiple values** (e.g., people with dual nationality). We currently pick a single value from a set based on frequency; 3. as noted, we applied dropout and oversampling mechanisms to reinforce **sparse attributes**, but these remain problematic; 4. it remains unclear **which aspects of the knowledge** are captured by our neural methods, especially by the EMB model whose embeddings abstract over the bits of knowledge. More insight is required to explain some differences we observed on individual facets.

Some of these limitations could possibly be alleviated by recent systems that aim to combine language models with background knowledge, such as K-BERT [65]. Applying such systems on our task is a viable future work direction.

9. Conclusions and future work

Despite their unique scarcity and non-redundancy of knowledge, lack of frequency/popularity priors, and potentially extreme ambiguity, the NIL entities have received surprisingly little attention in existing information extraction research.

This paper systematically investigated the role of explicit and implicit knowledge when determining identity of NIL entities mentioned in text. Given that the available information in text could be insufficient to establish identity, we enhanced it with *profiles*: background knowledge models that aim to capture implicit expectations left out in text, thus normalizing the comparison and accounting for the knowledge sparsity of most attributes. We tested 6 hypotheses about the role of different extent and quality of explicit knowledge, profiling enhancements, and reasoning methods. Imperfect attribute extraction led to lower performance in comparison to gold extraction, and profiling was able to fill certain gaps of the automatic extractors. As expected, higher ambiguity made the task much harder. However, the usefulness of profiling had no clear relation to the degree of data ambiguity.

The profiles are explicit representations, thus providing us with transparency and explanation on the identity decisions. We analyzed their behavior on two intrinsic experiments, thus testing another 5 hypotheses. Namely, we evaluated the profiling machines against instantial data in Wikidata, as well as against human judgments collected in a crowdsourcing task. Both experiments showed that the profiling methods get much closer to the typical or the correct value than the underlying baselines based on frequency or the Naive Bayes method. The prediction accuracy per attribute varies greatly, which can be largely explained through the notions of entropy, value size, and (number of) known attributes.

Further research should continue investigating how to best incorporate profiling components to fill the gaps in human communication. Our experiments revealed two key potential pitfalls, namely: 1. profiles built on top of incomplete or wrong information extracted from text can be misleading and counterproductive for the task of establishing long-tail identity 2. there is a notable discrepancy between the properties and their values found in text in comparison to those that can be learned from Wikidata. The role of other types of knowledge, i.e., intertextual and circumtextual, for determining identity of NILs should also be tested.

Future work needs to investigate how to apply EMB (the embeddings-based predictor) to the task of NIL clustering. In addition, the generalizability of our approaches to other entity types, e.g., organizations or locations, should be tested. This would entail automating of the property set selection for the AE system. Intuitively, an updated version of the EMB system that is able to compute entity embeddings on the fly would be natively applicable to any kind of entity without excessive manual engineering. Hence, we expect that an adapted EMB model will be able to generalize easier to different entity types. To better understand the strengths and weaknesses of our methods, it is also important to compare our methods to existing NIL clustering systems empirically.

With further understanding and engineering of these phenomena, we expect that such profiling machines would be able to natively address (at least) three standing challenges of modern-day IE: 1. scarcity of episodic knowledge, prominent both in knowledge bases and in communication; 2. unresolved ambiguity in communication, when the available knowledge is not necessarily scarce, yet prior expectations could lead to more reliable disambiguation; 3. anomaly detection, when a seemingly reliable machine interpretation is counter-intuitive and anomalous with respect to our expectations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Filip Ilievski: Conceptualization, Data curation, Formal analysis, Methodology, Investigation, Resources, Software, Writing - original draft, Writing - review & editing. **Eduard Hovy:** Conceptualization, Methodology, Project administration, Supervision, Writing - original draft. **Piek Vossen:** Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing - original draft. **Stefan Schlobach:** Methodology, Supervision, Writing - original draft. **Qizhe Xie:** Software.

Acknowledgments

The research reported in this paper has been funded by the Netherlands Organisation for Scientific Research (NWO) via the Spinoza fund.

References

[1] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion, in: Proceedings of the 20th ACM SIGKDD, ACM, 2014.

[2] H. Ji, R. Grishman, Knowledge base population: Successful approaches and challenges, in: Proceedings of ACL: Human Language Technologies-Volume 1, 2011.

[3] H. Ji, J. Nothman, B. Hachey, R. Florian, Overview of TAC-KBP2015 tri-lingual entity discovery and linking, in: Proceedings of the Eighth Text Analysis Conference (TAC2015), 2015.

[4] F. Ilievski, P. Vossen, S. Schlobach, Systematic study of long tail phenomena in entity linking, in: Proceedings of COLING, 2018, pp. 664–674.

[5] D. Kahneman, Thinking, Fast and Slow, Macmillan, 2011.

[20] We obtained *lifespan* and *century of birth* from birth and death dates.

[6] X. Cheng, D. Roth, Relational inference for wikification, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1787–1796.

[7] J. Daiber, M. Jakob, C. Hokamp, P.N. Mendes, Improving efficiency and accuracy in multilingual entity extraction, in: Proceedings of SEMANTiCS, 2013.

[8] A. Moro, A. Raganato, R. Navigli, Entity linking meets word sense disambiguation: a unified approach, TACL 2 (2014).

[9] D. Moussallem, R. Usbeck, M. Röeder, A.-C.N. Ngomo, MAG: A multilingual, knowledge-base agnostic and deterministic entity linking approach, in: Proceedings of the Knowledge Capture Conference, ACM, 2017.

[10] T.H. Nguyen, N. Fauceglia, M.R. Muro, O. Hassanzadeh, A.M. Gliozzo, M. Sadoghi, Joint learning of local and global features for entity linking via neural networks, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2310–2320.

[11] A. Sakor, I. Onando Mulang', K. Singh, S. Shekarpour, M. Esther Vidal, J. Lehmann, S. Auer, Old is gold: Linguistic driven approach for entity and relation linking of short text, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 2336–2346, http://dx.doi.org/10.18653/v1/N19-1243, https://www.aclweb.org/anthology/N19-1243.

[12] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph and text jointly embedding, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1591–1601.

[13] H. Zhong, J. Zhang, Z. Wang, H. Wan, Z. Chen, Aligning knowledge and text embeddings by entity descriptions, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 267–272.

[14] W. Fang, J. Zhang, D. Wang, Z. Chen, M. Li, Entity disambiguation by knowledge and text jointly embedding, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 260–269.

[15] I. Yamada, H. Shindo, H. Takeda, Y. Takefuji, Joint learning of the embedding of words and entities for named entity disambiguation, 2016, arXiv preprint arXiv:1601.01343.

[16] J.G. Moreno, R. Besançon, R. Beaumont, E. D'hondt, A.-L. Ligozat, S. Rosset, X. Tannier, B. Grau, Combining word and entity embeddings for entity linking, in: European Semantic Web Conference, Springer, 2017, pp. 337–352.

[17] L. Derczynski, K. Bontcheva, I. Roberts, Broad twitter corpus: A diverse named entity recognition resource, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1169–1179.

[18] J. Esquivel, D. Albakour, M. Martinez, D. Corney, S. Moussa, On the long-tail entities in news, in: European Conference on Information Retrieval, 2017.

[19] F. Ilievski, P. Vossen, M. Van Erp, Hunger for contextual knowledge and a road map to intelligent entity linking, in: International Conference on Language, Data and Knowledge, Springer, Cham, 2017, pp. 143–149.

[20] W. Radford, B. Hachey, M. Honnibal, J. Nothman, J.R. Curran, Naıve but effective NIL clustering baselines–CMCRC at TAC 2011, in: TAC, 2011.

[21] D. Graus, T. Kenter, M. Bron, E. Meij, M. De Rijke, et al., Context-based entity linking-university of Amsterdam at TAC 2012, in: TAC, 2012.

[22] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, A. Jung, Cross-lingual cross-document coreference with entity linking, in: TAC, 2011.

[23] Y. Hong, D. Lu, D. Yu, X. Pan, X. Wang, Y. Chen, L. Huang, H. Ji, Rpi blender tac-kbp2015 system description, in: Proc. Text Analysis Conference (TAC2015), 2015.

[24] I. Nagy, R. Farkas, Person attribute extraction from the textual parts of web pages, Acta Cybern. 20 (3) (2012) 419–440.

[25] B. Zhong, J. Liu, Y. Du, Y. Liaozheng, J. Pu, Extracting attributes of named entity from unstructured text with deep belief network, Int. J. Database Theory Appl. 9 (5) (2016) 187–196.

[26] G. Angeli, J. Tibshirani, J. Wu, C.D. Manning, Combining distant and partial supervision for relation extraction, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1556–1567.

[27] H. Adel, B. Roth, H. Schütze, Comparing convolutional neural networks to traditional models for slot filling, 2016, arXiv preprint arXiv:1603.05157.

[28] G. Ji, K. Liu, S. He, J. Zhao, Knowledge graph completion with adaptive sparse transfer matrix, in: AAAI, 2016, pp. 985–991.

[29] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: NIPS, 2013.

[30] Q. Xie, X. Ma, Z. Dai, E. Hovy, An Interpretable Knowledge Transfer Model for Knowledge Base Completion, Association for Computational Linguistics (ACL), 2017.

[31] K. Guu, J. Miller, P. Liang, Traversing knowledge graphs in vector space, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, ACL, 2015, pp. 318–327, http://aclweb.org/anthology/D15-1038.

[32] R. Socher, D. Chen, C.D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: NIPS, 2013.

[33] S. Riedel, L. Yao, A. McCallum, B.M. Marlin, Relation extraction with matrix factorization and universal schemas, in: HLT-NAACL, 2013, pp. 74–84.

[34] L. Yao, S. Riedel, A. McCallum, Universal schema for entity type prediction, in: Proceedings of the 2013 Workshop on Automated KB Construction, ACM, 2013, pp. 79–84.

[35] K. Lakshminarayan, S.A. Harp, T. Samad, Imputation of missing data in industrial databases, Appl. Intell. 11 (3) (1999) 259–275.

[36] R.K. Pearson, The problem of disguised missing data, ACM SIGKDD Explor. Newsl. 8 (1) (2006) 83–92.

[37] J.A. Abourbih, A. Bundy, F. McNeill, Using linked data for semi-automatic guesstimation, in: AAAI Spring Symposium: Linked Data Meets AI, 2010.

[38] M. Farid, I.F. Ilyas, S.E. Whang, C. Yu, LONLIES: estimating property values for long tail entities, in: Proceedings of SIGIR, ACM, 2016.

[39] M. Akbari, T.-S. Chua, Leveraging behavioral factorization and prior knowledge for community discovery and profiling, in: Proceedings of the ACM Conference on Web Search and Data Mining, 2017, pp. 71–79.

[40] D. Jurgens, That's what friends are for: Inferring location in online social media platforms based on social relationships, ICWSM 13 (13) (2013) 273–282.

[41] J. Mahmud, J. Nichols, C. Drews, Where is this tweet from? inferring home locations of twitter users, ICWSM 12 (2012) 511–514.

[42] D. Kahneman, A. Tversky, Prospect theory: An analysis of decision under risk, Econometrica (1979) 263–291.

[43] J. Pearl, Causality, Cambridge University Press, 2009.

[44] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, A. Kalai, Man is to computer programmer as woman is to homemaker? debiasing word embeddings, in: Advances in Neural Information Processing Systems, 2016, pp. 4349–4357.

[45] G. MacLachlan, I. Reid, Framing and Interpretation, Melbourne University Press, 1994.

[46] L. Jussim, J.T. Crawford, R.S. Rubinstein, Stereotype (in) accuracy in perceptions of groups and individuals, Curr. Dir. Psychol. Sci. (2015).

[47] J. Pujara, E. Augustine, L. Getoor, Sparsity and noise: Where knowledge graph embeddings fall short, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1751–1756.

[48] R.D. Ashmore, F.K. Del Boca, Conceptual approaches to stereotypes and stereotyping, Cogn. Process. Stereotyping Intergroup Behav. 1 (1981) 35.

[49] Z. Kunda, Social Cognition: Making Sense of People, MIT press, 1999.

[50] A.J. Dijker, W. Koomen, Stereotyping and attitudinal effects under time pressure, Eur. J. Soc. Psychol. 26 (1) (1996) 61–74.

[51] G. Stone, N. Gage, G. Leavitt, Two kinds of accuracy in predicting another's responses, J. Soc. Psychol. 45 (2) (1957) 245–254.

[52] A. Hopkinson, A. Gurdasani, D. Palfrey, A. Mittal, Demand-weighted completeness prediction for a knowledge base, 2018, arXiv preprint arXiv:1804.11109.

[53] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[54] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, JMLR (2014).

[55] J. Preiss, J. Dehdari, J. King, D. Mehay, Refining the most frequent sense baseline, in: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions, Association for Computational Linguistics, 2009, pp. 10–18.

[56] H. Zhang, The optimality of naive Bayes, AA 1 (2) (2004) 3.

[57] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: A CPU and GPU math compiler in Python, in: Proc. 9th Python in Science Conf, 2010, pp. 1–7.

[58] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of ICLR, 2014.

[59] M. Postma, F. Ilievski, P. Vossen, SemEval-2018 Task 5: Counting events and participants in the long tail, in: The SemEval-2018 Workshop, ACL, 2018.

[60] P. Vossen, F. Ilievski, M. Postma, R. Segers, Don't annotate, but validate: a data-to-text method for capturing event data, in: Proceedings of LREC, 2018.

[61] E. Pavlick, H. Ji, X. Pan, C. Callison-Burch, The gun violence database: A new task and data set for NLP, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2016, pp. 1018–1024, http://aclweb.org/anthology/D16-1106.

[62] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, vol. 96, 1996, pp. 226–231.

[63] K. Krippendorff, Content analysis. Beverly Hills, California: Sage Publications 7 (1980) l–84.

[64] S. Mohammad, G. Hirst, Distributional measures as proxies for semantic relatedness, CoRR abs/1203.1 (2012).

[65] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, P. Wang, K-bert: enabling language representation with knowledge graph, 2019, arXiv preprint arXiv:1909.07606.