Identifying Generic Noun Phrases

Nils Reiter and **Anette Frank**
Department of Computational Linguistics
Heidelberg University, Germany
{reiter,frank}@cl.uni-heidelberg.de

Abstract

This paper presents a supervised approach for identifying generic noun phrases in context. Generic statements express rule-like knowledge about kinds or events. Therefore, their identification is important for the automatic construction of knowledge bases. In particular, the distinction between generic and non-generic statements is crucial for the correct encoding of generic and instance-level information. Generic expressions have been studied extensively in formal semantics. Building on this work, we explore a corpus-based learning approach for identifying generic NPs, using selections of linguistically motivated features. Our results perform well above the baseline and existing prior work.

1 Introduction

Generic expressions come in two basic forms: generic noun phrases and generic sentences. Both express rule-like knowledge, but in different ways.

A **generic noun phrase** is a noun phrase that does not refer to a specific (set of) individual(s), but rather to a kind or class of individuals. Thus, the NP *The lion* in (1.a)[1] is understood as a reference to the class "lion" instead of a specific individual. Generic NPs are not restricted to occur with kind-related predicates as in (1.a). As seen in (1.b), they may equally well be combined with predicates that denote specific actions. In contrast to (1.a), the property defined by the verb phrase in (1.b) may hold of individual lions.

(1) a. The lion was the most widespread mammal.

b. Lions eat up to 30 kg in one sitting.

Generic sentences are characterising sentences that quantify over situations or events, expressing rule-like knowledge about habitual actions or situations (2.a). This is in contrast with sentences that refer to specific events and individuals, as in (2.b).

(2) a. After 1971 [Paul Erdős] also took amphetamines.

b. Paul Erdős was born [...] on March 26, 1913.

The genericity of an expression may arise from the generic (kind-referring, class-denoting) interpretation of the NP or the characterising interpretation of the sentence predicate. Both sources may concur in a single sentence, as illustrated in Table 1, where we have cross-classified the examples above according to the genericity of the NP and the sentence.

This classification is extremely difficult, because (i) the criteria for generic interpretation are far from being clear-cut and (ii) both sources of genericity may freely interact.

	S[gen+]	S[gen-]
NP[gen+]	(1.b)	(1.a)
NP[gen-]	(2.a)	(2.b)

Table 1: Generic NPs and generic sentences

The above classification of generic expressions is well established in traditional formal semantics (cf. Krifka et al. (1995))[2]. As we argue in this paper, these distinctions are relevant for semantic processing in computational linguistics, especially for information extraction and ontology learning and population tasks. With appropriate semantic analysis of generic statements, we can not only formally capture and exploit generic knowledge,

[1] All examples are taken from Wikipedia unless stated otherwise.

[2] The literature draws some finer distinctions including aspects like specificity, which we will ignore in this work.

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 40–49,
Uppsala, Sweden, 11-16 July 2010. ©2010 Association for Computational Linguistics

but also distinguish between information pertaining to individuals vs. classes. We will argue that the automatic identification of generic expressions should be cast as a machine learning problem instead of a rule-based approach, as there is (i) no transparent marking of genericity in English (as in most other European languages) and (ii) the phenomenon is highly context dependent.

In this paper, we build on insights from formal semantics to establish a corpus-based machine learning approach for the automatic classification of generic expressions. In principle our approach is applicable to the detection of both generic NPs and generic sentences, and in fact it would be highly desirable and possibly advantageous to cover both types of genericity simultaneously. Our current work is confined to generic NPs, as there are no corpora available at present that contain annotations for genericity at the sentence level.

The paper is organised as follows. Section 2 introduces generic expressions and motivates their relevance for knowledge acquisition and semantic processing tasks in computational linguistics. Section 3 reviews prior and related work. In section 4 we motivate the choice of feature sets for the automatic identification of generic NPs in context. Sections 5 and 6 present our experiments and results obtained for this task on the ACE-2 data set. Section 7 concludes.

2 Generic Expressions & their Relevance for Computational Linguistics

2.1 Interpretation of generic expressions

Generic NPs There are two contrasting views on how to formally interpret generic NPs. According to the first one, a generic NP involves a special form of **quantification**. Quine (1960), for example, proposes a universally quantified reading for generic NPs. This view is confronted with the most important problem of all quantification-based approaches, namely that the exact determination of the quantifier restriction (QR) is highly dependent on the context, as illustrated in (3)[3].

(3) a. Lions are mammals. QR: *all lions*

b. Mammals give birth to live young. QR: *less than half of all mammals*

c. Rats are bothersome to people. QR: *few rats*[4]

In view of this difficulty, several approaches restrict the quantification to only "relevant" (Declerck, 1991) or "normal" (Dahl, 1975) individuals.

According to the second view, generic noun phrases denote **kinds**. Following Carlson (1977), a kind can be considered as an individual that has properties on its own. On this view, the generic NP cannot be analysed as a quantifier over individuals pertaining to the kind. For some predicates, this is clearly marked. (1.a), for instance, attributes a property to the kind lion that cannot be attributed to individual lions.

Generic sentences are usually analysed using a special dyadic operator, as first proposed by Heim (1982). The dyadic operator relates two semantic constituents, the *restrictor* and the *matrix*:

$$Q[x_1,...,x_i](\underbrace{[x1,...,x_i]}_{\textit{Restrictor}};\underbrace{\exists y_1,...,y_i[x_1,..,x_i,y_1,...,y_i]}_{\textit{Matrix}})$$

By choosing GEN as a generic dyadic operator, it is possible to represent the two readings (a) and (b) of the characterising sentence (4) by variation in the specification of restrictor and matrix (Krifka et al., 1995).

(4) Typhoons arise in this part of the pacific.

(a) Typhoons in general have a common origin in this part of the pacific.

(b) There arise typhoons in this part of the pacific.

(a') GEN$[x;y]$(Typhoon(x);this-part-of-the-pacific$(y)\wedge$arise-in(x,y))

(b') GEN$[x;y]$(this-part-of-the-pacific(x);Typhoon$(y)\wedge$arise-in(y,x))

In order to cope with characterising sentences as in (2.a), we must allow the generic operator to quantify over situations or events, in this case, "normal" situations which were such that Erdős took amphetamines.

2.2 Relevance for computational linguistics

Knowledge acquisition The automatic acquisition of formal knowledge for computational applications is a major endeavour in current research

[3] Some of these examples are taken from Carlson (1977).

[4] Most rats are not even noticed by people.

and could lead to big improvements of semantics-based processing. Bos (2009), e.g., describes systems using automated deduction for language understanding tasks using formal knowledge.

There are manually built formal ontologies such as SUMO (Niles and Pease, 2001) or Cyc (Lenat, 1995) and linguistic ontologies like WordNet (Fellbaum, 1998) that capture linguistic and world knowledge to a certain extent. However, these resources either lack coverage or depth. Automatically constructed ontologies or taxonomies, on the other hand, are still of poor quality (Cimiano, 2006; Ponzetto and Strube, 2007).

Attempts to automatically induce knowledge bases from text or encyclopaedic sources are currently not concerned with the distinction between generic and non-generic expressions, concentrating mainly on factual knowledge. However, rule-like knowledge can be found in textual sources in the form of generic expressions[5].

In view of the properties of generic expressions discussed above, this lack of attention bears two types of risks. The first concerns the distinction between classes and instances, regarding the attribution of properties. The second concerns modelling exceptions in both representation and inferencing.

The distinction between **classes and instances** is a serious challenge even for the simplest methods in automatic ontology construction, e.g., Hearst (1992) patterns. The so-called IS-A patterns do not only identify subclasses, but also instances. *Shakespeare*, e.g., would be recognised as a hyponym of *author* in the same way as *temple* is recognised as a hyponym of *civic building*.

Such a missing distinction between classes and instances is problematic. First, there are predicates that can only attribute properties to a kind (1.a). Second, even for properties that in principle can be attributed to individuals of the class, this is highly dependent on the selection of the quantifier's restriction in context (3). In both cases, it holds that properties attributed to a class are not necessarily inherited by any or all instances pertaining to the class.

Zirn et al. (2008) are the first to present fully automatic, heuristic methods to distinguish between classes and instances in the Wikipedia taxonomy derived by Ponzetto and Strube (2007). They report an accuracy of 81.6% and 84.5% for different classification schemes. However, apart from a plural feature, all heuristics are tailored to specific properties of the Wikipedia resource.

Modelling exceptions is a cumbersome but necessary problem to be handled in ontology building, be it manually or by automatic means, and whether or not the genericity of knowledge is formalised explicitly. In artificial intelligence research, this area has been tackled for many years. Default reasoning (Reiter, 1980) is confronted with severe efficiency problems and therefore has not extended beyond experimental systems. However, the emerging paradigm of Answer Set Programming (ASP, Lifschitz (2008)) seems to be able to model exceptions efficiently. In ASP a given problem is cast as a logic program, and an answer set solver calculates all possible answer sets, where an answer set corresponds to a solution of the problem. Efficient answer set solvers have been proposed (Gelfond, 2007). Although ASP may provide us with very efficient reasoning systems, it is still necessary to distinguish and mark default rules explicitly (Lifschitz, 2002). Hence, the recognition of generic expressions is an important precondition for the adequate representation and processing of generic knowledge.

3 Prior Work

Suh (2006) applied a rule-based approach to automatically identify generic noun phrases. Suh used patterns based on part of speech tags that identify bare plural noun phrases, reporting a precision of 28.9% for generic entities, measured against an annotated corpus, the ACE 2005 (Ferro et al., 2005). Neither recall nor f-measure are reported. To our knowledge, this is the single prior work on the task of identifying generic NPs.

Next to the ACE corpus (described in more detail below), Herbelot and Copestake (2008) offer a study on annotating genericity in a corpus. Two annotators annotated 48 noun phrases from the British National Corpus for their genericity (and specificity) properties, obtaining a kappa value of 0.744. Herbelot and Copestake (2008) leave su-

[5] In the field of cognitive science, research on the acquisition of generic knowledge in humans has shown that adult speakers tend to use generic expressions very often when talking to children (Pappas and Gelman, 1998). We are not aware of any detailed assessment of the proportion of generic noun phrases in educational text genres or encyclopaedic resources like Wikipedia. Concerning generic sentences, Mathew and Katz (2009) report that 19.9% of the sentences in their annotated portion of the Penn Treebank are habitual (generic) and 80.1% episodic (non-generic).

pervised learning for the identification of generic expressions as future work.

Recent work by Mathew and Katz (2009) presents automatic classification of generic and non-generic sentences, yet restricted to habitual interpretations of generic sentences. They use a manually annotated part of the Penn TreeBank as training and evaluation set[6]. Using a selection of syntactic and semantic features operating mainly on the sentence level, they achieved precision between 81.2% and 84.3% and recall between 60.6% and 62.7% for the identification of habitual generic sentences.

4 Characterising Generic Expressions for Automatic Classification

4.1 Properties of generic expressions

Generic NPs come in various syntactic forms. These include definite and indefinite singular count nouns, bare plural count and singular and plural mass nouns as in (5.a-f). (5.f) shows a construction that makes the kind reading unambiguous. As Carlson (1977) observed, the generic reading of "well-established" kinds seems to be more prominent (g vs. h).

(5) a. The lion was the most widespread mammal.

b. A lioness is weaker [...] than a male.

c. Lions died out in northern Eurasia.

d. Metals are good conductors.

e. Metal is also used for heat sinks.

f. The zoo has one kind of tiger.

g. The Coke bottle has a narrow neck.

h. The green bottle has a narrow neck.

Apart from being all NPs, there is no obvious syntactic property that is shared by all examples. Similarly, generic sentences come in a range of syntactic forms (6).

(6) a. John walks to work.

b. John walked to work
(when he lived in California).

c. John will walk to work
(when he moves to California).

Although generic NPs and generic sentences can be combined freely (cf. Section 1; Table 1), both phenomena highly interact and quite often appear in the same sentence (Krifka et al., 1995). Also, genericity is highly dependent on contextual factors. Present tense, e.g., may be indicative for genericity, but with appropriate temporal modification, generic sentences may occur in past or future tense (6). Presence of a copular construction as in (5.a,b,d) may indicate a generic NP reading, but again we find generic NPs with event verbs, as in (5.e) or (1.b). Lexical semantic factors, such as the semantic type of the clause predicate (5.c,e), or "well-established" kinds (5.g) may favour a generic reading, but such lexical factors are difficult to capture in a rule-based setting.

In our view, these observations call for a corpus-based machine learning approach that is able to capture a variety of factors indicating genericity in combination and in context.

4.2 Feature set and feature classes

In Table 2 we give basic information about the individual features we investigate for identifying generic NPs. In the following, we will structure this feature space along two dimensions, distinguishing NP- and sentence-level factors as well as syntactic and semantic (including lexical semantic) factors. Table 3 displays the grouping into corresponding feature classes.

NP-level features are extracted from the local NP without consideration of the sentence context.

Sentence-level features are extracted from the clause (in which the NP appears), as well as sentential and non-sentential adjuncts of the clause. We also included the (dependency) relations between the target NP and its governing clause.

Syntactic features are extracted from a parse tree or shallow surface-level features. The feature set includes NP-local and global features.

Semantic features include semantic features abstracted from syntax, such as tense and aspect or type of modification, but also lexical semantic features such as word sense classes, sense granularity or verbal predicates.

Our aim is to determine indicators for genericity from combinations of these feature classes.

[6] The corpus has not been released.

Feature	Description
Number	sg, pl
Person	1, 2, 3
Countability	ambig, no noun, count, uncount
Noun Type	common, proper, pronoun
Determiner Type	def, indef, demon
Granularity	The number of edges in the WordNet hypernymy graph between the synset of the entity and a top node
Part of Speech	POS-tag (Penn TreeBank tagset; Marcus et al. (1993)) of the head of the phrase
Bare Plural	false, true
Sense[0-3]	WordNet sense. Sense[0] represents the sense of the head of the entity, Sense[1] its direct hypernym sense and so forth.
Sense[Top]	The top sense in the hypernym hierarchy (often referred to as "super sense")
Dependency Relation [0-4]	Dependency Relations. Relation[0] represents the relation between entity and its governor, Relation[1] the relation between the governor and its governor and so forth.
Embedded Predicate.Pred	Lemma of the head of the directly governing predicate of the entity
C.Tense	past, pres, fut
C.Progressive	false, true
C.Perfective	false, true
C.Mood	indicative, imperative, subjunctive
C.Passive	false, true
C.Temporal Modifier?	false, true
C.Number of Modifiers	numeric
C.Part of Speech	POS-tag (Penn TreeBank tagset; Marcus et al. (1993)) of the head of the phrase
C.Pred	Lemma of the head of the clause
C.Adjunct.Time	true, false
C.Adjunct.VType	main, copular
C.Adjunct.Adverbial Type	vpadv, sadv
C.Adjunct.Degree	positive, comparative, superlative
C.Adjunct.Pred	Lemma of the head of the adjunct of the clause
XLE.Quality	How complete is the parse by the XLE parser? fragmented, complete, no parse

Table 2: The features used in our system. C stands for the clause in which the noun phrase appears, "Embedding Predicate" its direct predicate. In most cases, we just give the value range, if necessary, we give descriptions. All features may have a NULL value.

	Syntactic	Semantic
NP-level	Number, Person, Part of Speech, Determiner Type, Bare Plural	Countability, Granularity, Sense[0-3, Top]
S-level	Clause.{Part of Speech, Passive, Number of Modifiers}, Dependency Relation[0-4], Clause.Adjunct.{Verbal Type, Adverbial Type}, XLE.Quality	Clause.{Tense, Progressive, Perfective, Mood, Pred, Has temporal Modifier}, Clause.Adjunct.{Time, Pred}, Embedded Predicate.Pred

Table 3: Feature classes

Name	Descriptions and Features
Set 1	Five best single features: Bare Plural, Person, Sense [0], Clause.Pred, Embedding Predicate.Pred
Set 2	Five best feature tuples: a. Number, Part of Speech b. Countability, Part of Speech c. Sense [0], Part of Speech d. Number, Countability e. Noun Type, Part of Speech
Set 3	Five best feature triples: a. Number, Clause.Tense, Part of Speech b. Number, Clause.Tense, Noun Type c. Number, Clause.Part of Speech, Part of Speech d. Number, Part of Speech, Noun Type e. Number, Clause.Part of Speech, Noun Type
Set 4	Features, that appear most often among the single, tuple and triple tests: Number, Noun Type, Part of Speech, Clause.Tense, Clause.Part of Speech, Clause.Pred, Embedding Predicate.Pred, Person, Sense [0], Sense [1], Sense[2]
Set 5	Features performing best in the ablation test: Number, Person, Clause.Part of Speech, Clause.Pred, Embedding Predicate.Pred, Clause.Tense, Determiner Type, Part of Speech, Bare Plural, Dependency Relation [2], Sense [0]

Table 4: Derived feature sets

5 Experiments

5.1 Dataset

As data set we are using the ACE-2 (Mitchell et al., 2003) corpus, a collection of newspaper texts annotated with entities marked for their genericity. In this version of the corpus, the classification of entities is a binary one.

Annotation guidelines The ACE-2 annotation guidelines describe generic NPs as referring to an arbitrary member of the set in question, rather than to a particular individual. Thus, a property attributed to a generic NP is in principle applicable to arbitrary members of the set (although not to all of them). The guidelines list several tests that are either local syntactic tests involving determiners or tests that cannot be operationalised as they involve world knowledge and context information.

The guidelines give a number of criteria to identify generic NPs referring to specific properties. These are (i) types of entities (*lions* in 3.a), (ii) suggested attributes of entities (*mammals* in 3.a), (iii) hypothetical entities (7) and (iv) generalisations across sets of entities (5.d).

(7) If a person steps over the line, they must be punished.

The general description of generic NPs as denoting arbitrary members of sets obviously does not capture kind-referring readings. However, the properties characterised (i) can be understood to admit kinds. Also, some illustrations in the guidelines explicitly characterise kind-referring NPs as generic. Thus, while at first sight the guidelines do not fully correspond to the characterisation of generics we find in the formal semantics literature, we argue that both characterisations have similar extensions, i.e., include largely overlapping sets of noun phrases. In fact, all of the examples for generic noun phrases presented in this paper would also be classified as generic according to the ACE-2 guidelines.

We also find annotated examples of generic NPs that are not discussed in the formal semantics literature (8.a), but that are well captured by the ACE-2 guidelines. However, there are also cases that are questionable (8.b).

(8) a. "It's probably not the perfect world, but you kind of have to deal with what you have to work with," he said.

b. Even more remarkable is the Internet, where information of all kinds is available about the government and the economy.

This shows that the annotation of generics is difficult, but also highlights the potential benefit of a corpus-driven approach that allows us to gather a wider range of realisations. This in turn can contribute to novel insights and discussion.

Data analysis A first investigation of the corpus shows that generic NPs are much less common than non-generic ones, at least in the newspaper genre at hand. Of the 40,106 annotated entities, only 5,303 (13.2%) are marked as generic. In order to control for bias effects in our classifier, we will experiment with two different training sets, a balanced and an unbalanced one.

5.2 Preprocessing

The texts have been (pre-)processed to add several layers of linguistic annotation (Table 5). We use MorphAdorner for sentence splitting and TreeTagger with the standard parameter files for part of speech tagging and lemmatisation. As we do not have a word sense disambiguation system available that outperforms the most frequent sense baseline, we simply used the most frequent sense (MFS). The countability information is taken from Celex. Parsing was done using the English LFG grammar (cf. Butt et al. (2002)) in the XLE parsing platform and the Stanford Parser.

Task	Tool
Sentence splitting	MorphAdorner [7]
POS, lemmatisation	TreeTagger (Schmid, 1994)
WSD	MFS (according to WordNet 3.0)
Countability	Celex (Baayen et al., 1996)
Parsing	XLE (Crouch et al., 2010)
	Stanford (Klein and Manning, 2003)

Table 5: Preprocessing pipeline

As the LFG-grammar produced full parses only for the sentences of 56% of the entities (partial parses: 37% of the entities), we chose to integrate the Stanford parser as a fallback. If we are unable to extract feature values from the f-structure produced by the XLE parser, we extract them from the Stanford Parser, if possible. Experimentation showed using the two parsers in tandem yields best results, compared to individual use.

[7] `http://morphadorner.northwestern.edu`

Feature Set			Generic			Non generic			Overall		
			P	R	F	P	R	F	P	R	F
		Baseline Majority	0	0	0	86.8	100	92.9	75.3	86.8	80.6
		Baseline *Person*	**60.5**	10.2	17.5	87.9	99.0	93.1	84.3	**87.2**	**85.7**
		Baseline Suh	28.9								
Feature Classes	Unbalanced	NP	31.7	56.6	40.7	92.5	81.4	86.6	84.5	78.2	81.2
		S	32.2	50.7	39.4	91.8	83.7	87.6	83.9	79.4	81.6
		NP/Syntactic	39.2	58.4	46.9	93.2	86.2	89.5	86.0	82.5	84.2
		S/Syntactic	31.9	22.1	26.1	88.7	92.8	90.7	81.2	83.5	82.3
		NP/Semantic	28.2	53.5	36.9	91.8	79.2	85	83.4	75.8	79.4
		S/Semantic	32.1	36.6	34.2	90.1	88.2	89.2	82.5	81.4	81.9
		Syntactic	**40.1**	66.6	**50.1**	94.3	84.8	89.3	**87.2**	82.4	84.7
		Semantic	34.5	56.0	42.7	92.6	83.8	88.0	84.9	80.1	82.4
		All	37.0	**72.1**	49.0	81.3	87.6	87.4	80.1	80.1	83.6
	Balanced	NP	30.1	71.0	42.2	94.4	74.8	83.5	85.9	74.3	79.7
		S	26.9	73.1	39.3	94.4	69.8	80.3	85.5	70.2	77.1
		NP/Syntactic	**35.4**	76.3	**48.4**	95.6	78.8	86.4	87.7	78.5	82.8
		S/Syntactic	23.1	77.1	35.6	94.6	61.0	74.2	85.1	63.1	72.5
		NP/Semantic	24.7	60.0	35.0	92.2	72.1	80.9	83.3	70.5	76.4
		S/Semantic	26.4	66.3	37.7	93.3	71.8	81.2	84.5	71.1	77.2
		Syntactic	30.8	**85.3**	45.3	96.9	70.8	81.9	88.2	72.8	79.7
		Semantic	30.1	67.5	41.6	93.9	76.1	84.1	85.5	75.0	79.9
		All	33.7	81.0	47.6	96.3	75.8	84.8	88.0	76.5	81.8
Feature Selection	Unbalanced	Set 1	**49.5**	37.4	42.6	90.8	94.2	92.5	85.3	86.7	86.0
		Set 2a	37.3	42.7	39.8	91.1	89.1	90.1	84.0	82.9	83.5
		Set 3a	42.6	54.1	47.7	92.7	88.9	90.8	86.1	84.3	85.2
		Set 4	42.7	**69.6**	52.9	94.9	85.8	90.1	88.0	83.6	85.7
		Set 5	45.7	64.8	**53.6**	94.3	88.3	91.2	87.9	85.2	**86.5**
	Balanced	Set 1	29.7	71.1	41.9	94.4	74.4	83.2	85.9	73.9	79.5
		Set 2a	36.5	70.5	48.1	94.8	81.3	87.5	87.1	79.8	83.3
		Set 3a	36.2	70.8	47.9	94.8	81.0	87.4	87.1	79.7	83.2
		Set 4	35.9	**83.1**	50.1	96.8	77.4	86.0	88.7	78.2	83.1
		Set 5	**37.0**	81.9	**51.0**	96.6	78.7	86.8	88.8	79.2	**83.7**

Table 6: Results of the classification, using different feature and training sets

5.3 Experimental setup

Given the unclear dependencies of features, we chose to use a Bayesian network. A Bayesian network represents the dependencies of random variables in a directed acyclic graph, where each node represents a random variable and each edge a dependency between variables. In fact, a number of feature selection tests uncovered feature dependencies (see below). We used the Weka (Witten and Frank, 2002) implementation BayesNet in all our experiments.

To control for bias effects, we created balanced data sets by oversampling the number of generic entities and simultaneously undersampling non-generic entities. This results in a dataset of 20,053 entities with approx. 10,000 entities for each class. All experiments are performed on balanced and unbalanced data sets using 10-fold cross-validation, where balancing has been performed for each training fold separately (if any).

Feature classes We performed evaluation runs for different combinations of feature sets: NP- vs. S-level features (with further distinction between syntactic and semantic NP-/S-level features), as well as overall syntactic vs. semantic features. This was done in order to determine the effect of different types of linguistic factors for the detection of genericity (cf. Table 3).

Feature selection We experimented with two methods for feature selection. Table 4 shows the resulting feature sets.

In **ablation testing**, a single feature in turn is temporarily omitted from the feature set. The feature whose omission causes the biggest drop in f-measure is set aside as a strong feature. This process is repeated until we are left with an empty feature set. From the ranked list of features f_1 to f_n we evaluate increasingly extended feature sets $f_1..f_i$ for i = 2..n. We select the feature set that yields the best balanced performance, at 45.7% precision and 53.6% f-measure. The features are given as Set 5 in Table 4.

As ablation testing does not uncover feature dependencies, we also experimented with **single, tuple** and **triple feature combinations** to determine features that perform well in combination. We ran evaluations using features in isolation and each possible pair and triple of features. We select the resulting five best features, tuples and triples of features. The respective feature sets are given as Set 1 to Set 3 in Table 4. The features that appear most often in Set 1 to Set 3 are grouped in Set 4.

Baseline Our results are evaluated against three baselines. Since the class distribution is unequal, a majority baseline consists in classifying each entity as non-generic. As a second baseline we chose the performance of the feature *Person*, as this feature gave the best performance in precision among those that are similarly easy to extract. Finally, we compare our results to (Suh, 2006).

6 Results and Discussion

The results of classification are summarised in Table 6. The columns Generic and Non-generic give the results for the respective class. Overall shows the weighted average of the classes.

Comparison to baselines Given the bias for non-generic NPs in the unbalanced data, the majority baseline achieves high performance overall (F: 80.6). Of course, it does not detect any generic NPs. The *Person*-based baseline also suffers from very low recall (R: 10.2%), but achieves the highest precision (P: 60.5 %). (Suh, 2006) reported only precision of the generic class, so we can only compare against this value (28.9 %). Most of the features and feature sets yield precision values above the results of Suh.

Feature classes, unbalanced data For the identification of generic NPs, syntactic features achieve the highest precision and recall (P: 40.1%, R: 66.6 %). Using syntactic features on the NP- or sentence-level only, however, leads to a drop in precision as well as recall. The recall achieved by syntactic features can be improved at the cost of precision by adding semantic features (R: 66.6 → 72.1, P: 40.1 → 37). Semantic features in separation perform lower than the syntactic ones, in terms of recall and precision.

Even though our results achieve a lower precision than the *Person* baseline, in terms of f-measure, we achieve a result of over 50%, which is almost three times the baseline.

Feature classes, balanced data Balancing the training data leads to a moderate drop in performance. All feature classes perform lower than on the unbalanced data set, yielding an increase in recall and a drop in precision. The overall performance differences between the balanced and unbalanced data for the best achieved values for the generic class are -4.7 (P), +13.2 (R) and -1.7 (F). This indicates that (i) the features prove to perform rather effectively, and (ii) the distributional bias in the data can be exploited in practical experiments, as long as the data distribution remains constant.

We observe that generally, the recall for the generic class improves for the balanced data. This is most noticeable for the S-level features with an increase of 55 (syntactic) and 29.7 (semantic). This could indicate that S-level features are useful for detecting genericity, but are too sparse in the non-oversampled data to become prominent. This holds especially for the lexical semantic features.

As a general conclusion, syntactic features prove most important in both setups. We also observe that the margin between syntactic and semantic features reduces in the balanced dataset, and that both NP- and S-level features contribute to classification performance, with NP-features generally outperforming the S-level features. This confirms our hypothesis that all feature classes contribute important information.

Feature selection While the above figures were obtained for the entire feature space, we now discuss the effects of feature selection both on performance and the distribution over feature classes. The results for each feature set are given in Table 6. In general, we find a behaviour similar to

	Syntactic	Semantic
NP	Number, Person, Part of Speech, Determiner Type, Bare Plural	Sense[0]
S	Clause.Part of Speech, Dependency Relation[2]	Clause.{Tense, Pred}

Table 7: Best performing features by feature class

the homogeneous classes, in that balanced training data increases recall at the cost of precision.

With respect to overall f-measure, the best single features are strong on the unbalanced data. They even yield a relatively high precision for the generic NPs (49.5%), the highest value among the selected feature sets. This, however, comes at the price of one of the lowest recalls. The best performing feature in terms of f-measure on both balanced and unbalanced data is Set 5 with Set 4 as a close follow-up. Set 5 achieves an f-score of 53.6 (unbalanced) and 51.0 (balanced). The highest recall is achieved using Set 4 (69.6% on the unbalanced and 83.1% on the balanced dataset). The results for Set 5 represent an improvement of 3.5 respectively 2.6 (unbalanced and balanced) over the best achieved results on homogeneous feature classes. In fact, Table 7 shows that these features, selected by ablation testing, distribute over all homogeneous classes.

We trained a decision tree to gain insights into the dependencies among these features. Figure 1 shows an excerpt of the obtained tree. The classifier learned to classify singular proper names as non-generic, while the genericity of singular nouns depends on their predicate. At this point, the classifier can correctly classify some of the NPs in (5) as kind-referring (given the training data contains predicates like “widespread”, “die out”, ...).

7 Conclusions and Future Work

This paper addresses a linguistic phenomenon that has been thoroughly studied in the formal semantics literature but only recently is starting to be addressed as a task in computational linguistics. We presented a data-driven machine learning approach for identifying generic NPs in context that in turn can be used to improve tasks such as knowledge acquisition and organisation. The classification of generic NPs has proven difficult even for humans. Therefore, a machine learning approach seemed promising, both for the identification of relevant features as for capturing contextual factors. We explored a range of features using homogeneous and mixed classes gained by alternative methods of feature selection. In terms of f-measure on the generic class, all feature sets performed above the baseline(s). In the overall classification, the selected sets perform above the majority and close to or above the *Person* baseline.

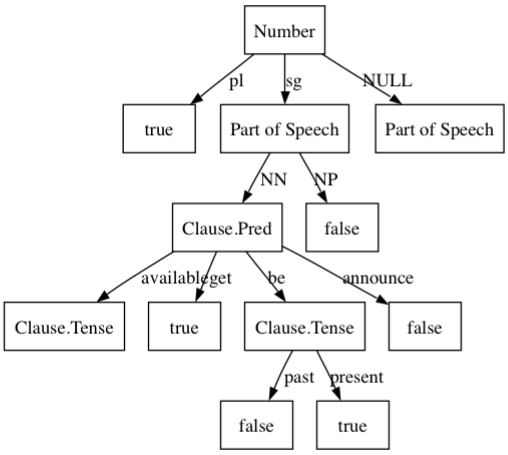

Figure 1: A decision tree trained on feature Set 5

The final feature set that we established characterises generic NPs as a phenomenon that exhibits both syntactic and semantic as well as sentence- and NP-level properties. Although our results are satisfying, in future work we will extend the range of features for further improvements. In particular, we will address lexical semantic features, as they tend to be effected by sparsity. As a next step, we will apply our approach to the classification of generic sentences. Treating both cases simultaneously could reveal insights into dependencies between them.

The classification of generic expressions is only a first step towards a full treatment of the challenges involved in their semantic processing. As discussed, this requires a contextually appropriate selection of the quantifier restriction[8], as well as determining inheritance of properties from classes to individuals and the formalisation of defaults.

References

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. CELEX2. Linguistic Data Consortium, Philadelphia.

Johan Bos. 2009. Applying automated deduction to natural language understanding. *Journal of Applied*

[8] Consider example (1.a), which is contextually restricted to a certain time and space.

Logic, 7(1):100 – 112. Special Issue: Empirically Successful Computerized Reasoning.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Marsuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of Grammar Engineering and Evaluation Workshop*.

Gregory Norman Carlson. 1977. *Reference to Kinds in English*. Ph.D. thesis, University of Massachusetts.

Philipp Cimiano. 2006. *Ontology Learning and Populating from Text*. Springer.

Dick Crouch, Mary Dalrymple, Ron Kaplan, Tracy King, John Maxwell, and Paula Newman, 2010. *XLE Documentation*. www2.parc.com/isl/groups/nltt/xle/doc/xle_toc.html.

Östen Dahl. 1975. On Generics. In Edward Keenan, editor, *Formal Semantics of Natural Language*, pages 99–111. Cambridge University Press, Cambridge.

Renaat Declerck. 1991. The Origins of Genericity. *Linguistics*, 29:79–102.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Lisa Ferro, Laurie Gerber, Janet Hitzeman, Elizabeth Lima, and Beth Sundheim. 2005. ACE English Training Data. Linguistic Data Consortium, Philadelphia.

Michael Gelfond. 2007. Answer sets. In *Handbook of Knowledge Representation*. Elsevier Science.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts, Amherst.

Aurelie Herbelot and Ann Copestake. 2008. Annotating genericity: How do humans decide? (a case study in ontology extraction). In Sam Featherston and Susanne Winkler, editors, *The Fruits of Empirical Linguistics*, volume 1. de Gruyter.

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Manfred Krifka, Francis Jeffry Pelletier, Gregory N. Carlson, Alice ter Meulen, Gennaro Chierchia, and Godehard Link. 1995. Genericity: An Introduction. In Gregory Norman Carlson and Francis Jeffry Pelletier, editors, *The Generic Book*. University of Chicago Press, Chicago.

Douglas B. Lenat. 1995. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38.

Vladimir Lifschitz. 2002. Answer set programming and plan generation. *Artificial Intelligence*, 138(1-2):39 – 54.

Vladimir Lifschitz. 2008. What is Answer Set Programming? In *Proceedings of AAAI*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330.

Thomas Mathew and Graham Katz. 2009. Supervised Categorization of Habitual and Episodic Sentences. In *Sixth Midwest Computational Linguistics Colloquium*. Bloomington, Indiana: Indiana University.

Alexis Mitchell, Stephanie Strassel, Mark Przybocki, JK Davis, George Doddington, Ralph Grishman, Adam Meyers, Ada Brunstein, Lisa Ferro, and Beth Sundheim. 2003. ACE-2 Version 1.0. Linguistic Data Consortium, Philadelphia.

Ian Niles and Adam Pease. 2001. Towards a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*.

Athina Pappas and Susan A. Gelman. 1998. Generic noun phrases in mother–child conversations. *Journal of Child Language*, 25(1):19–33.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the 22nd Conference on the Advancement of Artificial Intelligence*, pages 1440–1445, Vancouver, B.C., Canada, July.

Willard Van Orman Quine. 1960. *Word and Object*. MIT Press, Cambridge, Massachusetts.

Raymond Reiter. 1980. A logic for default reasoning. *Artificial Intelligence*, 13:81–132.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. *Proceedings of the conference on New Methods in Language Processing*, 12.

Sangweon Suh. 2006. Extracting Generic Statements for the Semantic Web. Master's thesis, University of Edinburgh.

Ian H. Witten and Eibe Frank. 2002. Data mining: practical machine learning tools and techniques with Java implementations. *ACM SIGMOD Record*, 31(1):76–77.

Cäcilia Zirn, Vivi Nastase, and Michael Strube. 2008. Distinguishing between instances and classes in the Wikipedia taxonomy. In *Proceedings of the 5th European Semantic Web Conference*.