Natural Language Engineering (2022), **28**, pp. 295–316
doi:10.1017/S1351324921000012

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Extracting possessions from text: Experiments and error analysis

Dhivya Chinnappa*† and Eduardo Blanco

University of North Texas, Denton, TX 76203, USA
*Corresponding author. E-mail: dhivyainfantchinnappa@my.unt.edu

(Received 15 November 2018; revised 2 January 2021; accepted 3 January 2021; first published online 9 March 2021)

Abstract

This paper presents a corpus and experiments to mine possession relations from text. Specifically, we target alienable and control possessions and assign temporal anchors indicating when a possession relation holds between the possessor and possessee. We work with intra-sentential possessor and possessees that satisfy lexical and syntactic constraints. We experiment with traditional classifiers and neural networks to automate the task. In addition, we analyze the factors that help to determine possession existence and possession type and common errors made by the best performing classifiers. Experimental results show that determining possession existence relies on the entire sentence, whereas determining possession type primarily relies on the verb, possessor and possessee.

Keywords: Possession; Temporal Processing; Corpus annotation

1. Introduction

Every language has a way of expressing possessive relationships (Aikhenvald and Dixon 2012). Possession is an asymmetric semantic relation between two entities, where one entity (the possessee) *belongs* to the other entity (the possessor) (Stassen 2009). When it comes to defining possession, *belongs* includes a wide range of relationships, including (hereafter, we use x to refer to the possessor and y to refer to the possessee) kinship (e.g., $[my]_x$ *oldest* $[son]_y$), part-whole (e.g., *the* $[car]_x$*'s* $[dashboard]_y$), physical and temporary possession (e.g., $[I]_x$ *have John's* $[book]_y$), possession of something intangible (e.g., $[John]_x$ *got the* $[flu]_y$ *last year*) and proximity (e.g., *The* $[shelf]_x$ *in the living room has a* $[glass\ sculpture]_y$).

Possession relations can be divided into alienable (also referred to as acquired, transferable, non-intimate, etc.) and inalienable (also referred to as inherent, inseparable, intimate, etc.). Possessees that can be separated from their possessors are alienable, and possessees that cannot normally be separated from their possessors are inalienable (Heine 1997). For example, $[John]_x$*'s* $[condo]_y$ is alienable, and $[John]_x$*'s* $[arm]_y$ is inalienable (some previous works would call the latter a part-whole relation instead Tratz and Hovy 2013). Control possession is defined as a relation in which the possessor has temporary access or control of the possessee (Tham 2004) but does not alienably possess it (e.g., $[John]_x$ *borrowed the* $[car]_y$ *for the weekend*). Following the aforecited works, possession goes beyond the ownership of property.

Possession relations can be expressed in a wide variety of syntactic constructions, including noun phrases (e.g., $[John]_x$*'s* $[car]_y$ only has two seats) and clauses (e.g., $[John]_x$ *bought a*

†Currently at Thomson Reuters.

© The Author(s), 2021. Published by Cambridge University Press.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

[*blue car*]$_y$ last week). The subject of a verb can map to either the possessor as exemplified above, or to the possessee (e.g., *The* [*car*]$_y$ *belongs to* [*John*]$_x$) (Aikhenvald and Dixon 2012).

Within computational linguistics, possession relations have usually been studied as part of larger studies that target any semantic relation between arguments connected with a rigid syntactic pattern (e.g., possessive constructions, nominals), as described in Section 2. Additionally, previous efforts have mostly targeted alienable possessions—or alternatively, ownership. The work presented here takes a different approach. We start by pairing people (plausible possessors) with concrete objects (plausible possessees). Then, we determine whether a possession relation exists, and if so, (a) determine the type (alienable or control) and (b) assign temporal anchors with respect to the event of which the possessor is the subject. We target all verbs, not only prototypical verbs of possession (e.g., *have*, *get*). Interestingly, our corpus includes several instances that do not contain *belong to* or other possession verbs. Thus, our approach extracts possessions intuitive to humans when there is no specific possession cue (e.g., we extract a control possession from *The* [*computer*]$_y$ *at work was slow*, [*I*]$_x$ *didn't get anything done*). The work presented here builds upon our previous work on possession extraction (Chinnappa and Blanco 2018a).

The main contributions of this paper are: (a) deterministic procedure to pair plausible possessors and possessees using lexical and syntactic constraints; (b) corpus annotating possession existence, possession type, and temporal anchors; (c) detailed corpus analysis including label distributions by verb and possessee; (d) experimental results with traditional learning and neural networks; and (e) model comparison and error analysis providing insights into the most salient sources of information to automate the task.

2. Previous work

Possession relations have been primarily studied as one of the many relations encoded in a given syntactic construction. For example, as many as 18 relations can be extracted from the English possessive (Tratz and Hovy 2013), including *temporal* (e.g., [*today*]$_x$*'s* [*rates*]$_y$), *extent* (e.g., [*6 hours'*]$_y$ [*drive*]$_x$), and *controller/owner/user* (one relation with three aliases, e.g., [*John*]$_x$*'s* [*car*]$_y$). The *controller/owner/user* relation is the closest relation to the alienable and control possessions we target in this paper. Another syntactic construction from which possession relations have been extracted is noun compounds (Tratz and Hovy 2010; Nakov and Hearst 2013), for example, [*family*]$_x$ [*estate*]$_y$. As a result of the syntactic constructions chosen by these previous works, they only target possession relations between possessors and possessees that appear nearby—either consecutive tokens or within a window of three tokens. Unlike these previous works, we do not only extract possession relations but also indicate the possession type (alienable or control) and temporally anchor the possession relation to indicate when it holds. Additionally, our approach is not restricted to (a) rigid syntactic patterns between possessor and possessee, or (b) possessors and possessees that appear nearby.

To the best of our knowledge, there is only one previous work on extracting possession relations without imposing syntactic constraints (Banea, Chen, and Mihalcea 2016). The authors work with personal blogs and determine whether possession relations hold between authors of blogs and concrete nouns mentioned in their blogs. Regarding time, they target possession relations at the time of the utterance (i.e., when the blog posts were published). Their definition of possession relations includes alienable and control possessions, but they do not distinguish between them. In a later work, the same authors present experiments to extract possessions from their blog corpus (Banea and Mihalcea 2018). They experiment with several machine learning algorithms like support vector machines, K-nearest neighbors, decision trees, feedforward neural networks, and Naive Bayes and report results. Unlike them, we distinguish between alienable and control possessions and assign temporal anchors with respect to an event in which the possessor participates.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

The work presented here expands our previous work on extracting possessions. In the first approach, we extract possessions at the sentence level (Chinnappa and Blanco 2018a). Later, we extract possessions from text and images from social media posts (Chinnappa, Murugan, and Blanco 2019). We have also explored possession attributes such as duration and co-possession (Chinnappa, Murugan, and Blanco 2020) and targeted changes in possession of important artworks over time (Chinnappa and Blanco 2018b; Chinnappa, Palmer, and Blanco 2020).

3. A corpus of possession relations

We create a corpus[a] following two steps. First, we generate intra-sentential pairs (x, y) of potential possessors (x) and possessees (y). Second, we annotate whether a possession exists, and if so, we also annotate the possession type and temporal anchors. Generating pairs a priori proved more effective than giving annotators plain text and asking them to annotate possessions.

We add our annotations on top of the English portion of OntoNotes (Hovy *et al.* 2006). Doing so has two advantages. First, OntoNotes contains texts from several domains and genres (e.g., conversational telephone speech, weblogs, and broadcast), thus we do not only work with newswire. Second, OntoNotes includes part-of-speech tags, named entities, and parse trees, three annotation layers that allow us to streamline the corpus creation process.

3.1 Pairing potential possessors and possessees

We generate pairs (x, y) such that it is plausible that x is the possessor of possessee y following these steps:

1. Collect as potential possessors all PERSON named entities and personal pronouns (part-of-speech tag PRP) *I*, *you*, *he*, *she*, *we* and *they*.
2. Discard potential possessors that are not the nominal subject (*nsubj* syntactic dependency) of a verb. Let us name that verb $verb_x$.
3. For each possessor, collect as potential possessees all nouns reachable from verb_x in the dependency tree and subsumed in WordNet (Miller 1995) by the synsets in Table 1.

Step 1 selects most people, as possessors are usually animate (Aikhenvald 2013). We do not consider possessive pronouns as they were targeted in previous work (Section 2). Additionally, our initial annotations revealed that possessive pronouns refer to plausible possessors selected in Step 1 after coreference resolution. Thus, this step helps to remove redundant possession relations. Step 2 reduces the number of potential possessors but note that we do not impose any restriction on verb_x, which may or may not be a verb of caused possession (Beavers 2011). Finally, Step 3 restricts the kind of objects considered as possessees. We intend to capture only physical possessions, and hence our synsets mostly include only concrete physical objects. Note that the selection of potential possessors and possessees uses syntactic constraints (Steps 2 and 3), but our constraints are less rigid than previous work (Section 2) and allow for distant possessor–possessee pairs.

We defined the list of synsets in Step 3 (Table 1) after manual examination of the WordNet hierarchy and prior to generating pairs. More specifically, we traversed the noun hierarchy following hyponymy relations from the root (i.e., *entity,n.01*) and selected synsets if all their children may be possessees. All synsets selected are subsumed by *artifact.n.01*, and all but *container.n.01*, *device.n.01*, *furnishing.n.02* and *vehicle.n.01* are direct hyponyms of *artifact.n.01*.

The total number of pairs generated after executing Steps (1–3) is 2025. In order to reduce the annotation effort, we set to annotate 1000 pairs. After trying several strategies, we reduce

[a] Available at https://github.com/dhivyainfant/possessionexistence18.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 1. WordNet synsets and their glosses used to select possessees (y) when generating pairs (x, y). *lemma.n.y* indicates the *y*th synset of *n*oun *lemma*

Synsets	Glosses
antiquity.n.03	An artifact surviving from the past
block.n.01	A solid piece of something (usually having flat rectangular sides)
cone.n.01	Any cone-shaped artifact
container.n.01	Any object that can be used to hold things (especially a large metal box-like object of standardized dimensions that can be loaded from one form of transport to another)
covering.n.02	An artifact that covers something else (usually to protect or shelter or conceal it)
decker.n.02	(often used in combinations) Something constructed with multiple levels
device.n.01	An instrumentality invented for a particular purpose
fabric.n.01	Artifact made by weaving or felting or knitting or crocheting natural or synthetic fibers
fixture.n.01	An object firmly fixed in place (especially in a household)
float.n.06	Something that floats on the surface of water
furnishing.n.02	(usually plural) The instrumentalities (furniture and appliances and other movable accessories including curtains and rugs) that make a home (or other area) livable
insert.n.02	An artifact that is inserted or is to be inserted
layer.n.01	Single thickness of usually some homogeneous substance
lemon.n.05	An artifact (especially an automobile) that is defective or unsatisfactory
marker.n.01	Some conspicuous object used to distinguish or mark something
plaything.n.01	An artifact designed to be played with
ready-made.n.01	A manufactured artifact (as a garment or piece of furniture) that is made in advance and available for purchase
squeaker.n.01	Any artifact that makes a squeaking sound when used
strip.n.02	artifact consisting of a narrow flat piece of material
vehicle.n.01	A conveyance that transports people or objects

the number of pairs as follows. First, we discard pairs with verb$_x$ *see*, *think*, *believe*, *say*, and *tell* because pilot annotations revealed that almost no possessions can be extracted (1757 pairs left). This observation is confirmed in the analysis of the final corpus: *communication* and *cognition* verbs have the lowest percentage of possession existence (Section 4). Second, we discard pairs (x, y) such that verb$_x$ occurs five or fewer times (979 pairs left). Table 2 presents basic counts per type of possessor (named entity or personal pronoun) and possessee (WordNet synset). We observe that most potential possessors are pronouns, and the vast majority of possessees are subsumed by *device.n.01*, *container.n.01*, *covering.n.02*, or *vehicle.n.01*. The category *other* refers to the synsets in Table 1 that are not mentioned in Table 2. Specifically *other* refers to synsets *marker.n.01*, *furnishing.n.02*, and *float.n.06*. Table 3 presents the distribution of the generated possessor–possessee pairs across the various domains in OntoNotes. We observe that the pairs are roughly uniformly distributed across five domains: newswire, broadcast news, broadcast conversation, web data, and magazine.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 2. Counts and percentages of pairs (x, y) generated per type of potential possessor (x) and possessee (y)

Possessee (y)	Possessor (x) is a					
is subsumed by	Pronoun	%	Person NE	%	All	%
device.n.01	295	38	74	36	369	38
container.n.01	189	24	30	15	219	22
covering.n.02	138	18	54	26	192	20
vehicle.n.01	124	16	36	18	160	16
fabric.n.01	6	1	7	3	13	1
block.n.01	9	1	2	1	11	1
plaything.n.01	3	0.5	2	1	5	0.5
fixture.n.01	4	0.5	0	0	4	0.5
antiquity.n.01	2	0.5	0	0	2	0.5
other	4	0.5	0	0	4	0.5
All	774	100	205	100	979	100

Table 3. Distribution of possessor–possessee pairs across various domains of OntoNotes

Domain	% possessor–possessee pairs
Newswire	22
Broadcast news	22
Broadcast conversation	20
Web data	17
Telephone conversation	1
Magazine	18

Figure 1 shows a sample sentence and the pairs generated. Note that these pairs include distant possessor–possessee pairs, not only subject–object pairs. Nouns *track*, *rest*, *Polaroid*, and *snapshot* are discarded as potential possessees because they are not subsumed by the synsets in Table 1.

We acknowledge that this process will not pair all possible possessors and possessees. For example, our potential possessors do not include nouns referring to a person like doctor, girl, or brother. Additionally, we restrict our plausible possessor to be a *nsubj* of verb_x thereby losing other plausible possessors in the sentence.

3.2 Annotating possession existence, possession type, and temporal anchors

After automatically generating pairs of potential possessors and possessees, annotators validate them manually. Annotations were done in-house, and the annotation interface showed the

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 4. Inter-annotator agreements (raw percentage and Cohen's κ)

Labels	%	κ
`yes`, `never`, `unk`, `inv`	86.1	0.79
`alienable`, `control`	82.5	0.77
`before yes`, `before no`	83.6	0.68
`during yes`, `during no`	88.8	0.75
`after yes`, `after no`	83.6	0.59

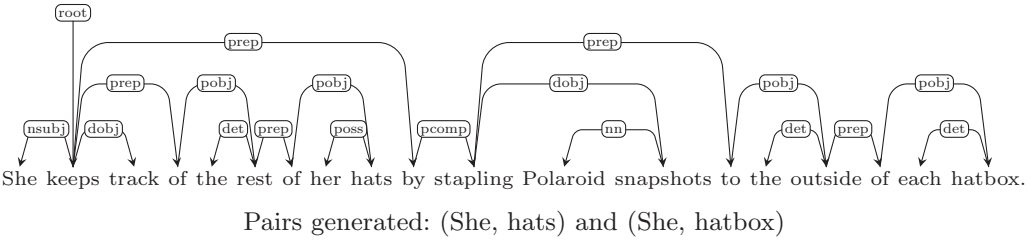


Figure 1. Sample sentence, dependency tree, and pairs (x, y) generated following the steps in Section 3.1.

current sentence (with the verb$_x$, potential possessor x, and potential possessee y highlighted), as well as the previous and next sentences.

The annotation process includes two major steps. First, annotators decide whether a possession relation exists between x and y based on the three sentences provided. More specifically, they choose from the following labels:

- `yes` if a possession exists at some point of time with respect to verb$_x$;
- `never` if a possession does not exist at any point of time with respect to verb$_x$;
- `unk` if it is sound to ask whether x is the possessor of possessee y, but there is not enough information to choose `yes` or `never`; and
- `inv` if either the potential possessor x is not animate, or the potential possessee y is nonsensical in the given context.

Second, annotators make two more annotations if the first label is `yes`:

- Possession type: whether the possession is `alienable` or `control`.
- Temporal anchors: whether the possession is true at some point of time `before`, `during`, and at some point of time `after` verb$_x$ (three binary decisions).

Following the literature (Tham 2004), we define *alienable* possession as a possessor owning a possessee, and *control* possession as a possessor having temporary access of a possessee but not ownership. Annotators were instructed to use world knowledge and interpret the sentences provided beyond what is explicitly stated. We present annotation examples in Section 3.3.

Inter-Annotator Agreement. Two computer science graduate students annotated the corpus. Both of them annotated 35% of all pairs (possession existence, possession type, and temporal anchors). We show inter-annotator agreements in Table 4. Cohen's κ for possession existence (labels `yes`, `never`, `unk` or `inv`) is 0.79. For possession type and temporal anchors, which are dependent on possession existence, we calculate inter-annotator agreement over the instances in which both annotators agree on possession existence. Cohen's κ for possession type (labels

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 5. Annotation examples for selected pairs (x, y). Recall that temporal anchors (B: before, D: during, and A: After) are annotated with respect to verb$_x$ and only if the main label is `yes`, that is, if there is an alienable or control possession relation between x and y

	Text (sentence of interest and context if relevant), pair (x, y) and verb$_x$	Label	B	D	A
1	But [he]$_x$ had extreme mood swings and [died]$_{verb_x}$ in a [car]$_y$ crash driving to work when I was five	`alienable`	✓	✓	X
2	After moving to the unoccupied zone, [wang]$_x$ began [carving]$_{verb_x}$ [seals]$_y$ in his spare time to support himself	`alienable`	X	X	✓
3	[He]$_y$ [kept]$_{verb_x}$ my father's [car]$_y$ for a year, without writing a confiscation order for it	`control`	X	✓	X
4	The horror and the heroics of the USS Cole, when the massive bomb ripped through the ship, were relived this evening by one of the crew, Liutenant Ann Chamberlain [I]$_x$ just [felt]$_{verb_x}$ a large—not so much a bang, but, to me, from where I was, it seemed like something had rammed the [ship]$_y$	`control`	✓	✓	✓
5	[LaToya]$_x$ also [described]$_{verb_x}$ being awakened in the night by Joseph wearing a "monster [mask]$_y$"	`never`		n/a	
6	[They]$_x$ [asked]$_{verb_x}$ him to come to them immediately because the reported [car]$_y$ had been seized	`unk`		n/a	
7	Will your political party straighten up and say, damn it, [we]$_x$ [have]$_{verb_x}$ to drop some of our ideological [baggage]$_y$?	`inv`		n/a	

`alienable` or `control`) is 0.77. Annotating whether the possession holds *before*, *during*, or *after* verb$_x$ obtains lower coefficients: 0.68, 0.75, and 0.59, respectively. Not surprisingly, the agreement for *during* is higher. Note that κ coefficients in the range 0.60–0.80 are considered substantial, and coefficients over 0.80 are usually considered nearly perfect (Artstein and Poesio 2008). Given these high agreements, the rest of pairs (65%) were annotated once.

3.3 Annotation examples

We present annotation examples using selected pairs of possessors and possessees in Table 5.

In Sentence (1), annotators interpreted that the relationship between *he* and *car* is an alienable possession. While not explicitly stated, annotators interpreted that *he* is an adult, and world knowledge tells us that most adults own the cars they drive unless a modifier indicates otherwise (e.g., *rental* car, my *father*'s car). Regarding temporal anchors, the possession between *he* and *car* is true before and during *died*, but not after.

Sentence (2) is a common example of alienable possession that is true after verb$_x$. The subject of a verb of creation (e.g., *make*, *build*) often becomes an alienable possessor of the direct object after the verb, but not before or during (because the object has not come into being yet).

Sentence (3) and (4) exemplify control possessions. In Sentence (3), *He* is borrowing *my father's car* for a period of time, and thus *He* has control over but does not own it. Regarding temporal anchors, nothing in the sentence indicates that *He* will have control over the car before or after *kept*. Note that our procedure to generate pairs would not generate the pair (*father*, *car*), but previous work has targeted possessives (Section 2).

In Sentence (4), verb$_x$ is *felt*, yet we extract a valid control possession. *I* is crew member of a warship and is describing his experience while on board. Annotators understood he had control possession of the ship (at least partially) before, during, and after, as *felt* did not last long and there is no indication that *I* left the boat immediately before or after *felt*.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

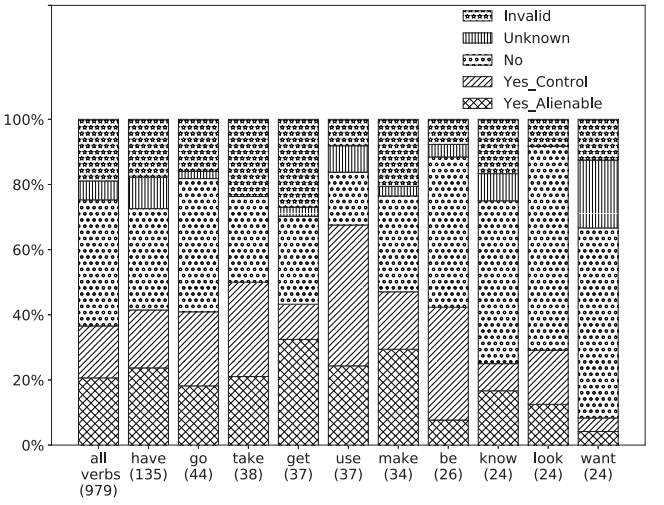


Figure 2. Label percentages for all verbs and the top 10 most frequent verbs of which the possessor is the subject. The numbers between parentheses under each verb indicate frequency.

Sentences (5–7) present examples in which annotators did not annotate a possession relation (labels `never`, `unk`, and `inv`). In Sentence (5), the *mask* belongs to *Joseph*. There is no indication that a possession relation exists between *LaToya* and *mask*, although *LaToya* was in close spatial proximity of the *mask* worn by *Joseph*. In Sentence (6), it is the case that *They* have some knowledge about the *car* that was seized, and it appears that *him*—not *They*—may be the alienable possessor. It is unclear, however, whether *They* and *car* are related by a control possession, thus annotators chose label `unk`. Finally, Sentence (7) exemplifies label `inv`. While *baggage* is most of the time a concrete object that passes the restrictions on potential possessees (Section 3), in this context, it is part of the metaphor *ideological baggage*. Since we only target concrete possessees, annotators chose `inv`.

4. Corpus analysis

Figure 2 presents label percentages for all verbs and the top 10 most frequent verbs. Overall, we extract possession relations from 36.5% of pairs (20.6%: `alienable`, 15.9%: `control`) and determine that no possession exists (label `never`) from 38.8% of pairs. Only 5.8% of pairs are annotated `unk`, and the higher percentage of `inv` (18.9%) is mostly due to potential possessees that can only be possessed in certain contexts, for example, compare $[\textit{They}]_x$ *asked* $[\textit{regulators}]_y$ *to suggest new ways to [. . .]* (label `inv`) versus $[\textit{They}]_x$ *replaced the* $[\textit{regulators}]_y$ *to control the flow of water* (label `yes`). When a possession holds, the most likely possession type is `alienable` (all verbs, 20.5% vs. 15.9%). We note, however, that `control` is the most likely possession type for many of the top 10 most frequent verbs: *go* (23% vs. 18%), *take* (29% vs. 21%), *use* (43% vs. 24%), *be* (35% vs. 8%), and *look* (17% vs. 12%). The possession type of the verb *want* is equally distributed between `alienable` and `control` (5% vs. 5%). Note that several verbs with high percentages of `alienable` and `control` labels are not prototypical verbs of possession (e.g., *go*, *use*, *know*).

Table 6 presents the distribution of possession types per temporal anchor with respect to verb_x (binary flags for before, during, and after). Alienable and control possessions show opposite trends for *before* and *after*, and substantially different distributions for *during*. The vast majority of control possessions are true during verb_x (85.3% vs. 14.7%), as well as a more modest majority of

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 6. Percentages of alienable and control POSSESSION relations annotated *yes* and *no* per temporal anchor (before, during, and after) with respect to verb$_x$

	Before		During		After	
	yes	*no*	*yes*	*no*	*yes*	*no*
Alienable	69.8	30.2	55.9	44.1	92.6	7.4
Control	28.8	71.2	85.3	14.7	33.3	66.7
All	52.0	48.0	68.7	31.3	67.8	32.2

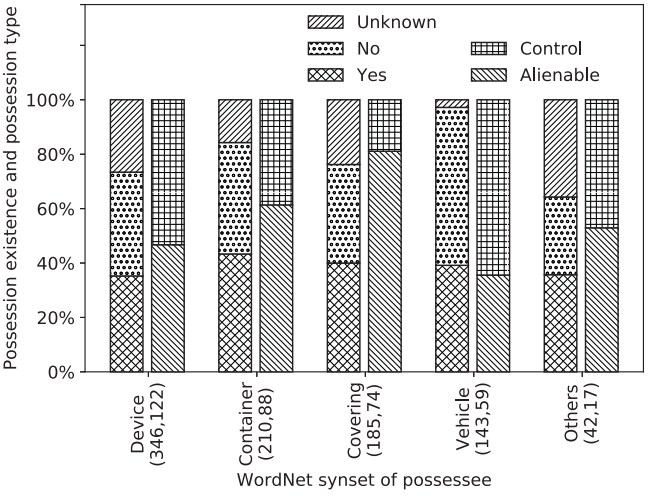


Figure 3. Distribution of possession existence and possession type per WordNet synset of the possessee. The numbers between parentheses indicate frequency of possession existence and possession type labels, respectively.

alienable possessions (55.9% vs. 44.1%). Alienable and control possessions, however, have opposite temporal anchors for *before* and *after*. Specifically, most alienable possessions are true before and after verb$_x$ (69.8% and 92.6%, respectively), and most control possessions are not true before and after verb$_x$ (71.2% and 66.7%).

Figure 3 presents the distribution of possession existence labels (labels `yes`, `never`, and `unk`) and possession type labels (labels `alienable` and `control`) per WordNet synset of the possessee. Regarding possession existence (Figure 3, left column for each synset), the lower percentage of `unk` (2%) with synset *vehicle.n.01* (car, plane, train, etc.) indicates that it is straightforward to decide whether a possession exists when the potential possessee is a vehicle (labels `yes`/`never` vs. `unk`). The *container.n.01* (bag, basket, bowl, etc.) also has a relatively low percentage of `unk` (16%). We note, however, the `yes` and `never` labels are almost uniformly distributed across all synsets, thus the synset of the possessee is insufficient to differentiate between these labels.

Regarding possession type (Figure 3, right columns for each synset), we observe that the WordNet synset (Miller 1995) *covering.n.02*, which mostly includes clothing items (jacket, blanket, etc.), is usually alienably possessed (81%). We also observe that possessees subsumed by *vehicle.n.01* (plane, aircraft, train, etc.) are involved in more control than alienable possessions (64% vs. 36%).

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

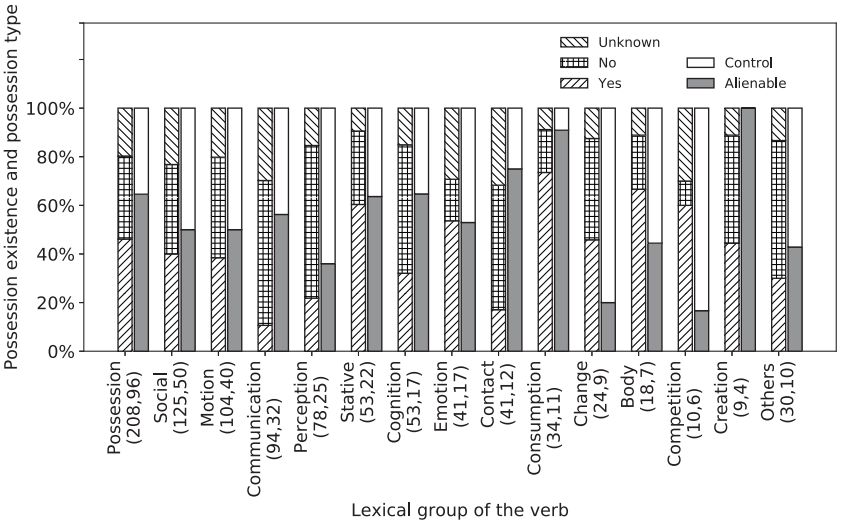


Figure 4. Distribution of possession existence and possession type over the lexical group of verb$_x$. The numbers between parentheses indicate frequency of possession existence and possession type labels, respectively.

Figure 4 presents the distribution of possession existence and possession type labels depending on the lexical group of verb$_x$. We obtain the lexical group of verb$_x$ from WordNet and consider the lexical group of the first sense of verb$_x$. Regarding possession existence (Figure 4, left column for each lexical group), we observe that the lexical group *possession* (have, own, buy, etc.) does not always indicate possession existence, although the percentage of `yes` label is substantial (46%). The lexical groups *perception* (hear, feel, etc.), *contact* (touch, hit, etc.), *cognition* (judge, doubt, etc.), and *communication* (ask, order, etc.) have the lowest percentages of `yes` (10–32%). These lower percentages support our decision to discard verbs *see*, *think*, *believe*, *say*, and *tell* during the pair generation process (Section 3), as they belong to the lexical groups *perception*, *cognition*, and *communication*. Labels `yes` and `never` are almost uniformly distributed with lexical groups *social* (do, help, make, etc.) and *motion* (walk, fly, swim, etc.), emphasizing that possession existence depends on context and cannot be determined exclusively based on the lexical group of verb$_x$. Other lexical groups such as *stative* (be, need, etc.), *consumption* (eat, drink, etc.), and *body* (brush, dress, etc.), which do not include prototypical verbs of possession, have high percentages of `yes` (46–74%).

Regarding possession type (Figure 4, right column for each lexical group), we observe that possessions involving a possessee from the *creation* lexical group (bake, make, create, etc.) are always alienable. The reason is that the creator of an artifact alienably possesses it after creating it. Similarly, the lexical group *change* (increase, move, etc.) has very high percentage of alienable possessions (91%). On the other hand, the lexical groups *communication* (ask, say, order, etc.) and *competition* (win, fight, etc.) have higher percentages of control possessions (83.3% vs. 16.7% and 80.0% vs. 20.0%, respectively).

Figure 5 presents the distribution of possession existence and possession type labels over the tense of verb$_x$. Simple past has higher percentage of `yes` label (47%) than other verb tenses. Other than that, verb tense does not help much to determine possession existence. Regarding possession type, it is interesting to note that *gerunds* have higher percentage of control possessions (63%), indicating that the possessor has temporal access of the possessee.

We do not present similar analyses with respect to temporal anchors (before, during and after) as we did not find interesting observations.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

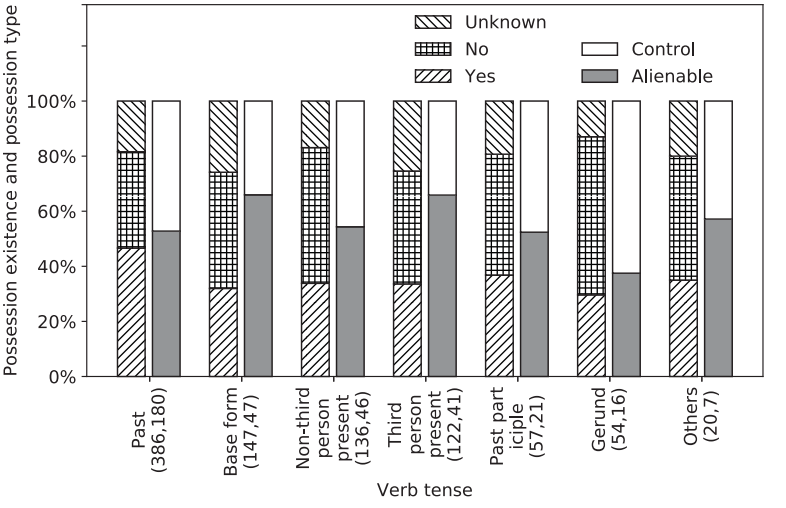


Figure 5. Distribution of possession existence and possession type over the tense of verb$_x$. The numbers between parentheses indicate frequency.

5. Experiments and results

Now that we have a corpus annotating possession relations, and we build classifiers to automate possession extraction. We conduct experiments using support vector machines and neural networks. Each pair (x, y) becomes an instance, and we create train (80%) and test (20%) sets with stratified label distribution. We report results using the test set after tuning hyperparameters using 10-fold cross-validation. More specifically, we train five classifiers and experiment with all instances but the ones annotated `inv`. The first classifier predicts possession existence (labels `yes`, `never`, or `unk`). The second classifier predicts possession types, that is, it classifies pairs between which a possession holds (label `yes`) into `alienable` or `control`. The third, fourth, and fifth classifiers predict temporal anchors, that is, theses classify pairs between which a possession holds—either alienable or control—into `before yes` or `before no`, `during yes` or `during no`, and `after yes` or `after no`.

5.1 Support vector machines

We trained the five classifiers using the support vector machine (SVM) implementation in scikit-learn (Pedregosa *et al.* 2011). We considered radial basis function kernel with C parameters in the range from 1 to 1,000,000 and γ parameters in the range 1×10^{-1} to 1×10^{-8}, and linear kernel with the same C parameter range. We chose the best model using grid search over the parameter ranges using 10-fold cross-validation. The features are summarized in Table 7. All features are extracted from the gold annotations of OntoNotes.

Verb features include the word and part-of-speech (POS) tag for the verb, previous and next tokens, as well as information regarding the outgoing and incoming dependencies. We also include a binary flag indicating whether the verb is a possession verb from an existing lexicon (Viberg 2010).

Possessor and *Possessee* features are very similar to *Verb* features, but we consider the concatenation of words and POS tags. *Possessee* features also include information derived from the WordNet hypernym paths to the root in the noun hierarchy, that is, *entity.n.01*, which subsumes all nouns in WordNet. More specifically, *WN synset* captures the synset from Figure 1 the possessee is subsumed by, and *WN path* are features capturing the top six synsets in the hypernym

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 7. Feature set used to extract possession relations (existence, type, and temporal anchors) with support vector machines

	Feature	Description
Verb	Word and tag	Word form and part-of-speech tag of verb
	is_possession_verb	Flag indicating whether *verb* is in the list of possession verbs
	Previous, next tokens	Word form and part-of-speech tags of the previous and next tokens
	dependency_out	Outgoing syntactic dependency type
	dependencies_in	Flags indicating the incoming syntactic dependencies
	Left, right children	Number of incoming syntactic dependencies to the left and right of *verb*
Possessor	Words	Concatenation of words
	Pos tags	Part-of-speech tag (full tag and first character, i.e., pronoun or noun)
	Previous, next tokens	Word form and part-of-speech tags of the previous and next tokens
	dependency_out	Outgoing syntactic dependency type
	dependencies_in	Flags indicating the incoming syntactic dependencies
Possessee	Same as possessor	Same features extracted for the possessor
	WN synset	WordNet synset from Table 1 the possessee is subsumed by
	WN path	WordNet synsets from entity.n.01 to the possessee
Syntactic paths	Possessor to verb	Syntactic path between possessor and verb
	Possessee to verb	Syntactic path between possessee and verb
	Possessor to possessee	Syntactic path between possessor and possessee

path from the possessee to *entity.n.01*. Finally, *Syntactic paths* features include three syntactic paths (syntactic dependency types and up/down symbols): from the possessor to the verb, from the possessee to the verb, and from the possessor to the possessee. The feature set is heavily inspired by many previous works (e.g, Gildea and Jurafsky 2002).

We experimented with SVM to establish a strong supervised baseline using linguistic information and to compare with neural networks that take as input only words along with information regarding verb_x, potential possessor x, and potential possessee y.

5.2 Neural networks

We experiment with feedforward and long short-term memory networks and use the implementations in Keras (Chollet *et al.* 2015) using TensorFlow backend (Abadi *et al.* 2015). All networks use GloVe embeddings with 100 dimensions (Pennington, Socher, and Manning 2014) and the Adam optimizer (Kingma and Ba 2014). Regarding input, we experiment with the potential possessor x, possessee y, verb_x, and the rest of the sentence. Note that verb_x, potential possessor x, and potential possessee y are known and the classifier takes as input any combination. The six architectures are depicted in Figure 6.

FFNN$_{verb}$. The first feedforward neural network takes as input the embedding of verb_x alone. It has a fully connected hidden layer with 50 neurons and uses softmax in the output layer of size 3 for predicting possession existence (labels `yes`, `never`, and `unk`) or size 2 for predicting possession type (labels `alienable` and `control`). Similarly, the output layer is of size 2 to predict the temporal anchors (labels `before yes` or `before no`, `during yes` or `during no`, and `after yes` or `after no`).

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

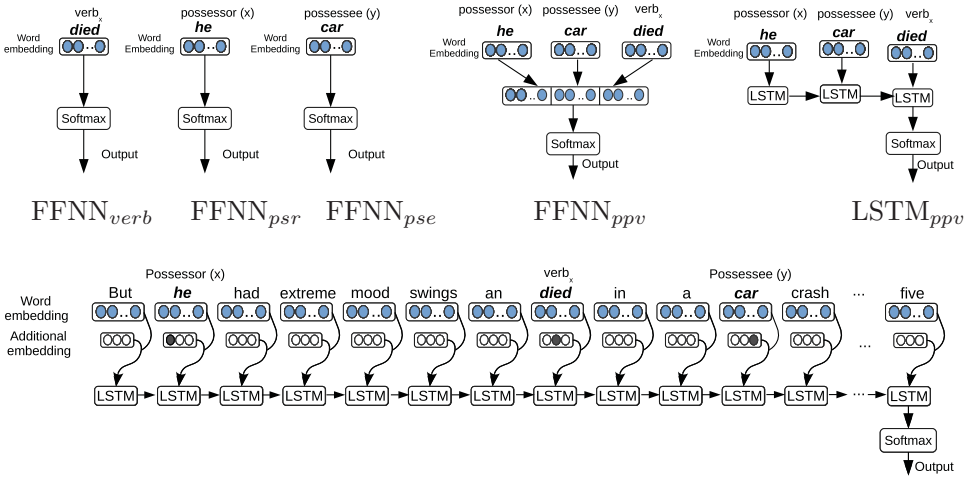


Figure 6. Neural network architectures. Feedforward neural networks take as input verb_x (FFNN_{verb}), the possessor (FFNN_{psr}), the possessee (FFNN_{pse}), or verb_x, the possessor, and the possessee (FFNN_{ppv}). The LSTMs take as input verb_x, the possessor, and the possessee (LSTM_{ppv}) and the sentence to which verb_x, possessor x, and possessee y belong (LSTM_{sent}). We exemplify the architectures with example 1 from Table 5, *But [he]$_x$ had extreme mood swings and [died]$_{\text{verb}_x}$ in a [car]$_y$ crash driving to work when I was five*.

FFNN$_{psr}$. This network takes as input the embedding of the potential possessor x and is otherwise identical to FFNN_{verb}.

FFNN$_{pse}$. This network takes as input the embedding of the potential possessee y and is otherwise identical to FFNN_{verb}.

FFNN$_{ppv}$. The FFNN_{ppv} architecture is similar to the previously discussed feedforward neural networks. The only difference is that it takes as input the word embeddings of the potential possessor x, the possessee y, and the verb_x.

LSTM$_{ppv}$. The first long short-term memory network takes as input a fixed-length sequence consisting of potential possessor x, possessee y, and verb_x. We used 100 long short term memory (LSTM) units and the output layer also uses softmax. While this LSTM has access to the same information as FFNN_{ppv}, we expect that the input, output, and forget gates will learn to update the cell state to better solve our task.

LSTM$_{sent}$. The second long short-term memory network takes as input the sequence of words from which the potential possessor x, possessee y, and verb_x were extracted. Specifically, we represent each element in the input with the concatenation of its word embedding and an additional one-hot embedding indicating if the token is verb_x, the potential possessor x, the potential possessee y, or none of them. LSTM_{sent} has access to the full sentence, and we expect that the memory update mechanism (i.e., the input, output, and forget gates) will learn the context most relevant for our task. Additionally, we expect that the additional embeddings will allow the LSTM to capture the right context surrounding verb_x, possessor x, and possessee y, as opposed to only model sentence meaning.

LSTM$_{noemb}$. The architecture and the input of the third long short-term memory network (not drawn in Figure 6) takes as input the sequence of words in the sentence from which verb_x, the potential possessor x, and the potential possessee y were extracted. The architecture is the same as

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 8. Results obtained using the majority baseline (possession existence: `never`, possession type: `alienable`), SVMs, and feedforward neural networks with single input ($FFNN_{verb}$, $FFNN_{psr}$, and $FFNN_{pse}$). An asterisk indicates statistical significance with respect to the majority baseline, and a dagger indicates statistical significance with respect to the best performing system ($LSTM_{sent}$ for possession existence and $LSTM_{ppv}$ for possession type shown in Table 9)

		Majority baseline			SVM			$FFNN_{verb}$			$FFNN_{psr}$			$FFNN_{pse}$		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Possession existence	`yes`	0.00	0.00	0.00	0.57	0.65	0.61	0.48	0.60	0.53	0.54	0.60	0.57	0.49	0.56	0.52
	`never`	0.41	1.0	0.58	0.61	0.53	0.56	0.61	0.55	0.58	0.52	0.70	0.60	0.58	0.57	0.57
	`unk`	0.00	0.00	0.00	0.44	0.43	0.44	0.44	0.43	0.44	0.31	0.22	0.25	0.50	0.05	0.10
	W. Avg.	0.17	0.41	0.24†	0.56	0.56	0.56*†	0.50	0.50	0.50*†	0.53	0.53	0.49*†	0.58	0.57	0.58*†
Possession type	`alienable`	0.56	1.0	0.72	0.64	0.57	0.61	0.64	0.75	0.69	0.57	0.65	0.60	0.67	0.75	0.71
	`control`	0.00	0.00	0.00	0.53	0.59	0.56	0.60	0.47	0.53	0.46	0.38	0.41	0.63	0.53	0.58
	W. Avg.	0.32	0.56	0.40†	0.59	0.58	0.58*	0.62	0.62	0.62*	0.52	0.53	0.52	0.65	0.65	0.65*

$LSTM_{sent}$ and the only difference is that we do not use additional embeddings to indicate $verb_x$, the potential possessor x, and possessee y.

5.3 Results

We present results in the three subtasks (possession existence, possession type, and temporal anchors) with the test set. To calculate statistical significance, we use two-tailed Z-tests (p-value < 0.01). We conducted all statistical significance tests over weighted F1 scores.

Possession Existence and Type. Tables 8 and 9 present results obtained with the majority baseline (possession existence: always `never`, possession type: always `alienable`), SVMs, and the neural networks. All models outperform the majority baseline in both tasks (possession existence F1: 0.24, possession type F1: 0.40), and most neural networks outperform SVMs. An asterisk (*) indicates statistical significance between each model and the majority baseline. All models obtain statistically better results predicting possession existence, and all but $FFNN_{psr}$ and $LSTM_{noemb}$ predicting possession types. In addition to the majority baseline, we also experimented with the majority baseline per verb, that is, a baseline that predicts the most frequent label per verb. This baseline obtains 0.51 F1 predicting possession existence, and 0.60 F1 predicting possession type (not shown in Tables 8 and 9).

Regarding possession existence (Tables 8 and 9, top blocks), the simple feedforward neural networks ($FFNN_{verb}$, $FFNN_{psr}$, $FFNN_{pse}$, and $FFNN_{ppv}$) perform similar to SVM (F1: 0.49–0.58 vs. 0.56), and $LSTM_{ppv}$ obtains surprisingly better results (F1: 0.69) although it takes as input the same information as $FFNN_{ppv}$. Despite the relatively small dataset (total: 979 instances), $LSTM_{sent}$ outperforms other networks (F1: 0.74), indicating that context beyond $verb_x$, possessor x, and possessee y are crucial to determine possession existence. The results also show that $LSTM_{sent}$ not only models sentences but also learns which context around $verb_x$, possessor x, and possessee y is most important to determine possession existence. Indeed, removing the additional embeddings for $verb_x$, the possessor x and possessee y results in 0.53 F1 (-32%, $LSTM_{noemb}$). The feedforward neural networks with single input ($FFNN_{verb}$, $FFNN_{psr}$, and $FFNN_{pse}$) obtain comparable F1 scores with `yes` and `never` labels and worse F1 scores with `unk`. On the other hand, $FFNN_{ppv}$ and $LSTM_{sent}$ obtain similar results across all labels, and $LSTM_{sent}$ obtains the best results across

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 9. Results obtained with $FFNN_{ppv}$, $LSTM_{ppv}$, $LSTM_{noemb}$, and $LSTM_{sent}$ for possession existence and possession type. An asterisk indicates statistical significance with respect to the majority baseline, and a dagger indicates statistical significance with respect to the best performing system ($LSTM_{sent}$ for possession existence and $LSTM_{ppv}$ for possession type)

		$FFNN_{ppv}$			$LSTM_{ppv}$			$LSTM_{noemb}$			$LSTM_{sent}$		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Possession existence	`yes`	0.54	0.53	0.54	0.65	0.69	0.67	0.54	0.53	0.53	0.74	0.72	0.73
	`never`	0.59	0.59	0.59	0.71	0.79	0.75	0.62	0.63	0.62	0.73	0.82	0.77
	`unk`	0.56	0.59	0.58	0.75	0.49	0.59	0.33	0.32	0.33	0.80	0.65	0.72
	W. Avg.	0.57	0.57	0.57*†	0.70	0.69	0.69*	0.53	0.53	0.53*†	0.75	0.75	0.74*
Possession type	`alienable`	0.68	0.75	0.71	0.67	0.80	0.73	0.58	0.78	0.67	0.67	0.60	0.63
	`control`	0.64	0.56	0.60	0.67	0.50	0.57	0.53	0.31	0.39	0.56	0.62	0.59
	W. Avg.	0.66	0.67	0.66*	0.66	0.67	0.67*	0.56	0.57	0.54	0.62	0.61	0.61*

all labels (F1: 0.72–0.77). The differences in results between $LSTM_{sent}$ are significant with respect to all other models except $LSTM_{ppv}$.

Regarding possession type (Tables 8 and 9, bottom blocks), all network architectures except $FFNN_{psr}$ (F1: 0.52) and $LSTM_{noemb}$ (F1: 0.54) yield better results than SVM (F1: 0.61–0.67 vs. 0.58). These results are intuitive, as possessors (in our corpus, either a personal pronoun or a PERSON named entity, and rarely the same individual) are not indicative of the possession type, but the possessee and $verb_x$ are indicative (Figures 3 and 4). The results also show that the possessee is the most important element to determine the possession type followed by $verb_x$: $FFNN_{pse}$ obtains 0.65 F1, $FFNN_{verb}$ 0.62 F1, and $FFNN_{psr}$ 0.52 F1. Indeed, the benefit of including the possessor and $verb_x$ is minimal ($FFNN_{pse}$: 0.65, $FFNN_{ppv}$: 0.66, and $LSTM_{ppv}$: 0.67), and including additional context is actually detrimental ($LSTM_{sent}$: 0.61). Despite having all the information available to the best performing model ($LSTM_{ppv}$), $LSTM_{sent}$ performs worse than $LSTM_{ppv}$. As the possessee and $verb_x$ alone are indicative of possession type, the presence of extra information in $LSTM_{sent}$ causes performance degradation. The differences in results between the best performing model ($LSTM_{ppv}$) and other models except the majority baseline are not statistically significant.

Temporal Anchors. Table 10 presents results obtained with SVMs and the neural networks that outperform the majority baseline in this subtask. $FFNN_{psr}$ and $FFNN_{pse}$, the only models that do not have access to $verb_x$, always predict the majority label (*yes* for all temporal anchors, Table 6).

$FFNN_{verb}$, which only has access to the verb, is the best performing system (before: 0.74, during: 0.76, and after: 0.74). SVM virtually obtains the same results as $FFNN_{verb}$ (before: 0.76, +2.7%; during: 0.72, −5.2%; and after: 0.73, −1.3%). Having access to the possessor and possessee ($FFNN_{ppv}$ and $LSTM_{ppv}$) and the full sentence ($LSTM_{noemb}$ and $LSTM_{sent}$) decreases results. As expected, F1 scores are higher with the most frequent label: `yes` is more frequent than `never` with all temporal anchors, especially with *during* and *after* (Table 6). These results show that $verb_x$ is critical to determine temporal anchors, and that the possessor and possessee are unnecessary.

6. Model comparison and error analysis

In this section, we compare the results obtained with selected neural networks and provide insights into the errors they make. We select pairs of networks such that the second network yields better results than the first one. Additionally, we discuss the errors made by the best performing

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Table 10. Results obtained when predicting temporal anchors with respect to verb_x. We show results with SVM and the neural networks that outperform the majority baseline. An asterisk indicates statistical significance with respect to the majority baseline

		Before			During			After		
		P	R	F1	P	R	F1	P	R	F1
SVM	Yes	0.76	0.82	0.78	0.78	0.82	0.80	0.78	0.86	0.82
	No	0.77	0.71	0.74	0.57	0.52	0.55	0.61	0.48	0.54
	W. Avg.	0.76	0.76	0.76*	0.72	0.72	0.72	0.72	0.74	0.73*
FFNN_{verb}	Yes	0.73	0.79	0.76	0.78	0.94	0.85	0.78	0.88	0.83
	No	0.74	0.68	0.71	0.77	0.43	0.56	0.65	0.48	0.55
	W. Avg.	0.74	0.74	0.74*	0.78	0.78	0.76	0.74	0.75	0.74*
FFNN_{ppv}	Yes	0.62	0.61	0.61	0.78	0.80	0.79	0.77	0.73	0.75
	No	0.57	0.59	0.58	0.55	0.52	0.53	0.48	0.52	0.50
	W. Avg.	0.60	0.60	0.60*	0.71	0.71	0.71	0.67	0.67	0.67
LSTM_{ppv}	Yes	0.71	0.76	0.73	0.80	0.84	0.82	0.79	0.76	0.77
	No	0.71	0.65	0.68	0.62	0.57	0.59	0.52	0.57	0.54
	W. Avg.	0.71	0.71	0.71*	0.74	0.75	0.75	0.70	0.69	0.70*
LSTM_{noemb}	Yes	0.66	0.66	0.66	0.76	0.69	0.72	0.80	0.73	0.77
	No	0.62	0.62	0.62	0.44	0.52	0.48	0.52	0.61	0.56
	W. Avg.	0.64	0.64	0.64*	0.66	0.64	0.65	0.71	0.69	0.70*
LSTM_{sent}	Yes	0.62	0.95	0.75	0.72	0.96	0.82	0.71	0.84	0.77
	No	0.86	0.35	0.50	0.71	0.22	0.33	0.43	0.26	0.32
	W. Avg.	0.73	0.67	0.63*	0.72	0.72	0.67	0.62	0.65	0.63

networks to shed light on what is needed to obtain better results. The discussion centers around determining possession existence labels (labels `yes`, `never` and `unk`) and possession type labels (labels `alienable` and `control`), as determining temporal anchors benefits almost exclusively from verb_x.

6.1 Possession existence

FFNN_{verb} versus FFNN_{ppv}. While a feedforward neural network with verb_x alone is a strong baseline (FFNN_{verb} F1: 0.50), including the possessor and possessee brings a 14% relative improvement (FFNN_{ppv} F1: 0.57). FFNN_{ppv} primarily fixes errors made by FFNN_{verb} in which the possessee must be taken into account for determining possession existence, as exemplified below:

1. $[\text{We}]_x$ $[\text{have}]_{\text{verb}_x}$ a video of that $[\text{plane}]_y$. (Gold: `never`)
2. $[\text{They}]_x$ have $[\text{received}]_{\text{verb}_x}$ $[\text{weapons}]_y$ from Israel under past peace accords. (Gold: `yes`)

In the first example, FFNN_{verb} predicts *yes*, as it only has access to verb_x (i.e., *have*), which is almost always annotated `yes` in the training corpus. Taking into account the possessee (i.e., *planes*) allows FFNN_{ppv} to correctly predict `never`. We note that FFNN_{ppv} would also predict `never` given $[\text{We}]_x$ $[\text{have}]_{\text{verb}_x}$ *a* $[\text{plane}]_y$, which is wrong. This is not surprising, as FFNN_{ppv} is far from obtaining perfect results, and additional context (LSTM_{sent}) is beneficial.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

In the second example, FFNN_{verb} wrongfully predicts `unk` despite *received* being a common verb of transfer of possession. Having access to the possessee (i.e., *weapons*) is beneficial again, and FFNN_{ppv} correctly predicts `yes`.

FFNN_{ppv} versus LSTM_{ppv}. While both FFNN_{ppv} and LSTM_{ppv} have the same input (embeddings for possessor x, possessee y, and verb_x), LSTM_{ppv} outperforms FFNN_{ppv} by a substantial margin (F1: 0.57 vs. 0.69, +21%). We credit the higher F1 score of LSTM_{ppv} to the internal cell state. Specifically, LSTM_{ppv} remembers different information from verb_x, the possessor, and possessee based on the current input element as well as the cell state obtained from previous elements in the sequence.

LSTM_{ppv} predicts the following examples correctly while FFNN_{ppv} does not

3. As soon as police officers pulled out the $[\text{gun}]_y$ and held it up, $[I]_x$ $[\text{knew}]_{verb_x}$ it was my rifle. (Gold: `yes`)
4. Though almost everybody at the playing site had been looking for the 26-year-old Soviet to beat the Pennsylvania-based computer, $[\text{he}]_x$ $[\text{gave}]_{verb_x}$ the $[\text{machine}]_y$ a far worse drubbing than many expected. (Gold: `yes`)

FFNN_{ppv} erroneously predicts `unk` and `never`, respectively, in example 3 and example 4. Note that the third example is a dialog snippet, the person talking is the owner of a gun shop and *gun* corefers with *rifle*. The gold label of the fourth example is `yes` because a control possession exists between the player (i.e., *he*, *the 26-year-old Soviet*) and the machine he is playing against.

LSTM_{ppv} versus LSTM_{sent}. In addition to possessor x, possessee y, and verb_x, having access to the full sentence is beneficial to predict possession existence (LSTM_{ppv} F1: 0.69; LSTM_{sent} F1: 0.74, +7%).

The following examples are correctly predicted by LSTM_{sent} but incorrectly predicted by LSTM_{ppv}:

5. Hounded by terror, I drove another kilometre and then $[\text{I}]_x$ $[\text{stopped}]_{verb_x}$ to make sure the $[\text{car}]_y$ wasn't on fire. (Gold: `yes`)
6. $[\text{Dog Chapman}]_x$ $[\text{was}]_{verb_x}$ in the same $[\text{truck}]_y$ you saw. (Gold: `yes`) LSTM_{ppv} incorrectly predicts `never` in both examples. Regarding the fifth example, we hypothesize that having access to the full sentence allows LSTM_{sent} to determine that the *car* is driven by *I*, thus a possession holds despite verb_x is *stopped*. It is harder to justify why LSTM_{sent} correctly predicts the sixth example. We hypothesize that LSTM_{sent} determines that *Dog Chapman*, *was*, and *truck* are in the same clause, thus a possession holds despite verb *was* is not a prototypical verb of possession.

LSTM_{sent} versus Gold. LSTM_{sent} is the best performing system predicting possession existence but obtains an overall F1 score of 0.74 thus is far from making perfect predictions. We identified four recurrent phenomena in the errors made by LSTM_{sent}.

First, a better model will need to reveal entailments involving commonsense and world knowledge. Consider the example below:

7. At one o'clock at night he came to the captain and the lieutenant colonel and they told him that $[\text{I}]_x$ was $[\text{wearing}]_{verb_x}$ a green $[\text{garment}]_y$. (Gold: `yes`)

A possession relation always exists between clothes and the people wearing them, and LSTM_{sent} is unable to determine so. The most common errors (35%) made by LSTM_{sent} are due to its inability to identify commonsense entailments. We believe this issue could be alleviated with a larger corpus of possessions or incorporating external resources (Liu and Singh 2004). Second, sentences with two or more pronouns are often incorrectly predicted. For example, LSTM_{sent} predicts `yes` in the following example:

8. But $[\text{we}]_x$ still $[\text{hear}]_{verb_x}$ him moaning at night because the Navy has a few $[\text{ships}]_y$ left, and to satisfy him the Navy's sea lift forces were given to a new Air Force bureaucracy in Illinois, its space operations to another command in Colorado, the frogmen to a new Army

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

bureaucracy in Fort Bragg, and the Navy's Indian Ocean and Persian Gulf forces to an Army bureaucracy in Florida. (Gold: `never`)

We believe that LSTM_{sent} fails in part because it is unable to understand the entities to which pronouns *we* and *him* refer to, and in particular, that they do not corefer with *the Navy*. This error type constitutes 29% of all errors. Third, LSTM_{sent} almost always outputs the wrong label when negation is present, as exemplified below:

9. But $[\text{I}]_x$ do not $[\text{have}]_{verb_x}$ a $[\text{computer}]_y$. (Gold: `never`)

Even with negations as simple as the one above, LSTM_{sent} fails. Failure to understand negation in a sentence causes 8% of all errors. We believe incorporating models to determine the scope (Fancellu, Lopez, and Webber 2016) and focus (Sarabi and Blanco 2017) of negation may help avoiding this issue. Fourth, determining event factuality (Sauri and Pustejovsky 2009) is required to determine possession existence, and LSTM_{sent} does not explicitly attempt to do so. Consider the example below, which LSTM_{sent} incorrectly predicts `yes`:

10. Maybe before the weapon and then later he said, Oh, $[\text{I}]_x$ should have $[\text{got}]_{verb_x}$ the nuclear $[\text{weapon}]_y$ before I invaded. (Gold: `never`)

Determining that *have got* is not factual (i.e., it did not occur) is key to determine that a possession relation does not exist between *I* and *weapon*. This phenomenon contributes to 6% of the errors.

6.2 Possession type

FFNN_{verb} versus FFNN_{ppv}. While verb_x alone is a strong baseline for determining possession existence (FFNN_{verb} F1: 0.62), including the possessor and possessee brings a 6.5% improvement (FFNN_{ppv} F1: 0.66). In the following examples, verb_x is a prototypical verb of possession, but the possessee is key to determine possession type:

11. $[\text{We}]_x$ will $[\text{give}]_{verb_x}$ to the head of transition for the President-elect, this $[\text{key}]_y$. (Gold: `control`)
12. $[\text{I}]_x$ $[\text{have}]_{verb_x}$ a bunch of $[\text{vehicles}]_y$ and only use the appropriate one for the task at hand. (Gold: `alienable`)

In example 11, *We* has a control possession of the key, as *We* are the temporary keepers until it is given to the President-elect. While in principle someone may have an alienable possession of a *key*, FFNN_{ppv} learnt from the training corpus that it is a control possession. In example 12, knowing that the possessee is a *vehicle* allows FFNN_{ppv} to determine that is is an alienable possession.

LSTM_{sent} versus LSTM_{ppv}. The experimental results show that the most useful information to determine possession type is verb_x, the possessor, and possessee. Indeed, having access to the full sentence is detrimental (LSTM_{ppv} F1: 0.67, LSTM_{sent} F1: 0.61, −9%).

The following examples are incorrectly predicted as alienable possessions by LSTM_{sent} but correctly predicted as control possessions by LSTM_{ppv}:

13. $[\text{They (Police)}]_x$ $[\text{showed}]_{verb_x}$ them my father's car and they testified that it was not the $[\text{car}]_y$ they saw, so they released me. (Gold: `control`)
14. $[\text{We}]_x$ have $[\text{got}]_{verb_x}$ people signing up, volunteers coming on, telephones being put in and $[\text{computers}]_y$, so we will be fully operational within the next day or two. (Gold: `control`)

In example 13, the *Police* showed the *car* to some people thus they have control possession of the *car*. LSTM_{ppv} successfully predicts control possession, and we believe that the main reason is that the possessor is *they* and the verb is *showed*. On the other hand, LSTM_{sent} has access to the full sentence and most likely makes a mistake because of the presence of a possessive construction (i.e., *father's car*) and the fact that coreference is not modeled. In example 14, *We* has control possession of the *computers*. LSTM_{ppv} successfully predicts control possession, but LSTM_{sent} wrongfully predicts alienable possession.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

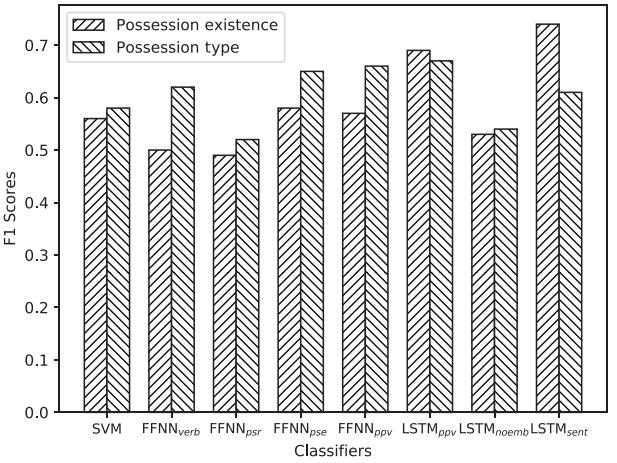


Figure 7. F1 scores obtained with several models to determine possession existence and possession type.

LSTM$_{ppv}$ versus Gold. The best classifier predicting possession type obtains relatively modest results in the task of determining possession existence (LSTM$_{ppv}$ F1: 0.67). We conducted an error analysis and identified two main sources of errors.

First, world knowledge that is not explicitly stated in the sentence or larger context is often necessary to determine possession type, as it is to determine possession existence. Consider the following example:

15. We're also checking out other developments right now, including this story that's just coming in from the Associated Press right now.

    [I]$_x$ just [want]$_{verb_x}$ to get it from our [computers]$_y$. (Gold: `control`) World knowledge tells us that employees (in this example, a journalist) have control possession of the computers they work with, not alienable possession. Lack of world knowledge is the most common cause (68%) of errors made by LSTM$_{ppv}$.

Second, in order to better predict possession type, a system must reveal presuppositions, that is, implicit assumptions about the world or background beliefs whose truth is taken for granted. Consider the example below:

16. [They]$_x$'re [trying]$_{verb_x}$ to keep [planes]$_y$ flying on days they'd normally park them, says Roger Bard, president of Mr. Mitchell Travel Service in Burnsville, N.C. (Gold: `control`)

The sentence above has at least the following presuppositions: *They fly planes* (because they are trying to keep planes flying on certain days) and *They have control possession of the planes* (because they fly and park planes). Inability to capture presuppositions contributes to 23% of errors made by LSTM$_{ppv}$.

Based on the error analysis, we attribute the errors made by the best performing model to the difficulty of the possession extraction task, and the inability of neural networks to capture linguistic phenomena. As discussed in Section 4, possessions are not always explicit in text. World knowledge plays a vital role in determining possessions. As the word embeddings used in this task are not generated from possession corpora, it is likely that they do not capture world knowledge containing possession information. Additionally, several studies (Qin *et al.* 2018; Sohrab and Miwa 2018; Bender and Koller 2020) have shown that neural networks fail to capture the underlying semantic information in texts.

Figure 7 summarizes the weighted F1 scores for determining possession existence and possession type. In general, we observe that determining possession existence is easier than possession type, and that determining existence benefits from context (LSTM$_{sent}$) while determining type does not.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

7. Conclusions

Possession relations are present in all languages, and they can reflect relationships, values, concepts, and cultural changes (Aikhenvald 2013). In this paper, we extract possession relations from text enforcing lexical and syntactic constraints. Specifically, we extract alienable and control possessions and specify temporal anchors with respect to the verb of which the possessor is the subject.

We have presented a corpus annotating possession existence, possession types, and temporal anchors following two steps. First, we automatically pair potential possessors and possessees, resulting in 979 pairs. Second, we manually validate pairs by annotating possession existence (labels `yes`, `never`, `unk`, and `inv`), possession type (labels `alienable` or `control`), and temporal anchor (labels *before* `yes` or `no`, *during* `yes` or `no`, and *after* `yes` or `no`). Inter-annotator Cohen's κ coefficients show that the annotation task can be done reliably (Table 4). Our corpus includes several instances which contain verbs that are not canonically though of as *belong to* possession. A corpus analysis (Section 4) shows that many possessions can be extracted from *social* and *motion* verbs, which are not prototypical verbs of possession. Additionally, the analysis shows that some verbs and possessees are more likely to have certain possession types. For example, *coverings* are usually part of alienable possessions, and *communication* verbs are usually part of control possessions.

Experimental results show that the task can be automated to some extent. Although neural networks fail to capture the underlying semantics, they outperform SVMs trained with features extracted from linguistic structure, although we experiment with a relatively small dataset. Determining possession existence benefits from having access to the full sentence, but determining possession type does not. Finally, the best performing models to determine temporal anchors rely only on verb$_x$. We expect the models to perform better with access to more data.

Beyond fundamental research, extracting possessions has several potential applications. Many applications such as clinical support systems, recommendation systems, and intelligent personal assistants would benefit from extracting possessions. Recommendation systems could work better if they were aware of a person's physical possessions. For instance, people who own a house may be interested in household items. For historical artifacts of great value (e.g., paintings like Mona Lisa and Guernica, which are worth millions of dollars), it is important to know the possessors, the transfer of possession, and their possession timeline to prove their provenance. Detecting the absence of possession is also useful. A physically challenged person could benefit from a possession extraction system, which could detect the absence of a necessary equipment like a wheel chair and notify a volunteer organization. This could be especially beneficial in underdeveloped countries with limited resources. Mining possession types also be useful in several cases. For example, marketers may target people who do not alienably possess something, and certain skills may be inferred from the kind of objects people have control possession over. For example, people having a control possession of an 18-wheeler most likely know how to drive large trucks and have a commercial driver's license.

References

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mané D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., ViÉgas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y. and Zheng X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

Aikhenvald A.Y. (2013). Possession and ownership: a cross-linguistic perspective. In Aikhenvald A.Y. and Dixon R.M.W. (eds), *Possession and Ownership: A Cross-Linguistic Typology*, Chapter 1. Oxford: Oxford University Press, pp. 1–64.

Aikhenvald, A.Y. and Dixon R.M.W. (2012). *Possession and Ownership: A Cross-Linguistic Typology*. *Explorations in Linguistic Typology*. Oxford: Oxford University Press.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Artstein R. and Poesio M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics* **34**(4), 555–596.

Banea C., Chen X. and Mihalcea R. (2016). Building a dataset for possessions identification in text. In Calzolar N., Choukri K., Declerck T., Goggi S., Grobelnik M., Maegaard B., Mariani J., Mazo H., Moreno A., Odijk J. and Piperidis S. (eds), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).

Banea C. and Mihalcea R. (2018). Possession identification in text. *Natural Language Engineering* **24**(04), 1–22.

Beavers J. (2011). An aspectual analysis of ditransitive verbs of caused possession in English. *Journal of Semantics* **28**(1), 1–54.

Bender E.M. and Koller A. (2020). Climbing towards NLU: on meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Online: Association for Computational Linguistics, pp. 5185–5198.

Chinnappa D. and Blanco E. (2018a). Mining possessions: existence, type and temporal anchors. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. New Orleans, Louisiana, USA: Association for Computational Linguistics, pp. 496–505.

Chinnappa D. and Blanco E. (2018b). Possessors change over time: a case study with artworks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2278–2287.

Chinnappa D., Murugan S. and Blanco E. (2019). Extracting possessions from social media: images complement language. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 663–672.

Chinnappa D., Murugan S. and Blanco E. (2020). Beyond possession existence: duration and co-possession. In *Proceedings of the 58th Annual Meeting of the* Association for Computational Linguistics. Online: Association for Computational Linguistics, pp. 8332–8341.

Chinnappa D., Palmer A. and Blanco E. (2020). WikiPossessions: possession timeline generation as an evaluation benchmark for machine reading comprehension of long texts. In *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 1110–1117.

Chollet F., et al. (2015). *Keras*. Available at https://github.com/fchollet/keras.

Fancellu F., Lopez A. and Webber B. (2016). Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 495–504.

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

Heine B. (1997). *Possession: Cognitive Sources, Forces, and Grammaticalization*. Cambridge *Studies in Linguistics*. Cambridge: Cambridge University Press.

Hovy E., Marcus M., Palmer M., Ramshaw L. and Weischedel R. (2006). OntoNotes: the 90% solution. In *NAACL'06: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*. Morristown, NJ, USA: Association for Computational Linguistics, pp. 57–60.

Kingma D.P. and Ba J. (2014). Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980 .

Liu H. and Singh P. (2004). ConceptNet—a practical commonsense reasoning tool-kit. *BT Technology Journal* **22**(4), 211–226.

Miller G.A. (1995). WordNet: a lexical database for English. *In Communications of the ACM*, vol. **38**, pp. 39–41.

Nakov P.I. and Hearst M.A. (2013). Semantic interpretation of noun compounds using verbal and other paraphrases. *ACM Transactions on Speech and Language* **10**(3), 13:1–13:51.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M. and Duchesnay E. (2011). Scikit-learn: machine learning in python. *Journal of Machine Learning Research* **12**, 2825–2830.

Pennington J., Socher R. and Manning C.D. (2014). GloVe: global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Qin G., Yao J.-G., Wang X., Wang J. and Lin C.-Y. (2018). Learning latent semantic annotations for grounding natural language to structured data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 3761–3771.

Sarabi Z. and Blanco E. (2017). If no media were allowed inside the venue, was anybody allowed? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume* 1, *Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 860–869.

Sauri R. and Pustejovsky J. (2009). FactBank: a corpus annotated with event factuality. *Language Resources and Evaluation* **43**(3), 227–268.

Sohrab M.G. and Miwa M. (2018). Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2843–2849.

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press

Stassen L. (2009). *Predicative Possession*. Oxford Studies in Typology and Linguistic Theory. Oxford/New York: Oxford University Press.

Tham S.W. (2004). *Representing Possessive Predication: Semantic Dimensions and Pragmatic Bases*. PhD Thesis, Stanford University.

Tratz S. and Hovy E. (2010). A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL'10. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 678–687.

Tratz S. and Hovy E.H. (2013). Automatic interpretation of the English possessive. In *ACL (1)*. The Association for Computer Linguistics, pp. 372–381.

Viberg A. (2010). Basic verbs of possession. *CogniTextes* **4**.

Cite this article: Chinnappa D and Blanco E (2022). Extracting possessions from text: Experiments and error analysis. *Natural Language Engineering* **28**, 295–316. https://doi.org/10.1017/S1351324921000012

https://doi.org/10.1017/S1351324921000012 Published online by Cambridge University Press