Relation Extraction for the Food Domain without Labeled Training Data – Is Distant Supervision the Best Solution?

Melanie Reiplinger, Michael Wiegand, and Dietrich Klakow

Spoken Language Systems, Saarland University, D-66123 Germany
{melanie.reiplinger|michael.wiegand|dietrich.klakow}@lsv.uni-saarland.de

Abstract. We examine the task of relation extraction in the food domain by employing distant supervision. We focus on the extraction of two relations that are not only relevant to product recommendation in the food domain, but that also have significance in other domains, such as the fashion or electronics domain. In order to select suitable training data, we investigate various degrees of freedom. We consider three processing levels being argument level, sentence level and feature level. As external resources, we employ manually created surface patterns and semantic types on all these levels. We also explore in how far rule-based methods employing the same information are competitive.

1 Introduction

In view of the large interest in food in many parts of the population and the ever increasing amount of new dishes, there is a need of automatic knowledge acquisition, especially relation extraction. Some relations, such as food items that can be served together or relations that express that two food items can be substituted by each other, are highly relevant to product recommendation systems or virtual customer advice. Unfortunately, such knowledge does not exist in conventional (structured) knowledge bases, yet it has been found that domain-specific corpora, i.e. unstructured natural language texts, contain an abundance of such relations [1].

In this paper, we focus on distant supervision [2] for relation extraction, where training data (i.e. sentences with mentions of particular relation instances) are automatically generated with the help of a relation database. Such a database contains argument pairs representing instances for relations that one wants to extract (e.g. <*rice pudding*, *fruit salad*> is an entity pair that expresses the relation that the two food items can be served together). Distant supervision rests on the assumption that sentences with mentions of those entity pairs are genuine instances of the relation (Sentence (1)). This way of producing labeled training data (given a relation database) is considerably faster than manually labeling sentences from a corpus in which food items occur. Of course, this approximation is not guaranteed to produce correct mentions of relation instances as exemplified in Sentence (2), so one also needs to devise methods to remove spurious relation mentions.

(1) For tonight, I planned to have rice pudding with fruit salad.
(2) Other types of food I like are pizza, falafel, rice pudding and fruit salad.

The aim of this investigation is to examine the different degrees of freedom in the design of a relation extraction classifier for the food domain based on distant supervision. Among the different aspects of such a classifier, we consider different kinds of knowledge (semantic types and surface patterns) and apply them on different processing levels. We also want to show that one has to take into consideration special properties of the relations to be extracted. For different relations, different kinds of classifier configurations may be suitable. We also want to critically assess whether predictive types of knowledge sources actually require a distantly supervised classifier, or whether a simple incorporation of such knowledge into rule-based classification already produces comparable results.

Our experiments are carried out on German data, but our findings should carry over to other languages since the issues we address are language universal. For general accessibility, all examples are given as English translations.

2 Data & Annotation

The relations that are to be extracted are *SubstitutedBy* and *SuitsTo* as illustrated in Table 1. We focus on these two relations because we consider them not only relevant to customer advice/product recommendation in the food domain, but also to similar applications in other types of domains. Customers want to know which items suit together, be it two food items that can be used as a meal, two fashion items that can be worn together, or two electronic devices that can be combined/connected in some way. Substitutes are relevant in all situations in which item A is out of stock but item B can be offered as an alternative.

As a gold standard, we randomly extracted from a domain-specific corpus about 2200 sentences in which at least two food items co-occur and manually labeled them with their pertaining relation. As a corpus, we chose a crawl of *chefkoch.de* [3], the largest German web platform dealing with food-related issues. This corpus also serves as an unlabeled dataset from which training data for distant supervision are to be extracted. In addition to the two relations from above, we introduce the label *Other* for cases in which either another relation between the target food items is expressed, or where their co-occurrence is coincidental. The class distribution in our gold standard is also shown in Table 1 (last column). On a subset of 400 sentences, an interannotation agreement of $\kappa = 0.78$ was measured which can be considered *substantial* [4].

For distant supervision, a relation database of entity tuples representing different relation instances is required. We make use of the resource introduced in [5] that has been specifically designed for distant supervision experiments. Table 2 lists the size of this database[1] together with the corresponding amount of sentences in our corpus where these food pairs match. By *None*, we understand all pairs that are neither contained in *SuitsTo* nor in *SubstitutedBy*.

[1] We excluded any food pairs from our manually labeled test set (i.e. gold standard).

The resource was created by giving two annotators partially instantiated relations, such as *SuitsTo(broccoli,?)* or *SubstitutedBy(beef roulade,?)*. The annotators then produced lists of food items that fit those relations, e.g. {*potatoes*, *fillet of pork*, *mushrooms*, ...} (food items that suit to *broccoli*) and {*goulash*, *braised meat*, *rolled pork*, ...} (food items that can be substituted by *beef roulade*). The annotators were allowed to consult various information sources for research, such as the internet. However, in order to obtain unbiased results, they were specifically asked not to focus on a particular source, e.g. a particular website. Note that the resource also contains other relations than the ones considered in this paper (for instance, relations that describe for what event a particular food item is suited, or which food items are recommended/not recommended for people with a particular health condition). It would be beyond the scope of this paper to examine all those different relations for distant supervision. We focus on the two relations due to their relevance to other domains (see discussion above).

Table 2 shows that there is a huge number of potentially negative data. However, one should keep in mind that such pairs may also contain positive pairs, since our relation database (i.e. the pairs representing instances of *SuitsTo* and *SubstitutedBy*) is not exhaustive.

Table 1. The different relations and their distribution in our gold standard.

Relation	**Description**	**Example**	**Perc.**
SuitsTo	food items that are typically consumed together	My kids love the simple combination of fish fingers with mashed potatoes.	60
SubstitutedBy	similar food items commonly consumed in the same situations	We usually buy margarine instead of butter.	9
Other	other relation *or* co-occurrence of food items is co-incidental	On my shopping list, I've got bread, cauliflower, ...	31

Table 2. Coverage of the relation database on the unlabeled food corpus.

Relation	**Argument Pairs**	**Matched Sentences**
SuitsTo	1,374	44,692
SubstitutedBy	781	34,771
None	62,191	1,187,101

3 Method

Figure 1 presents the most important aspects of the relation extraction system examined in this paper. The figure can be read in the following way. There are two main *knowledge* sources that are examined in this paper being *patterns* and *types*. The sources can be harnessed by either of the two classifiers, *distant supervision* and *rule-based classification*. With regard to distant supervision, there

are also three different processing levels to be considered. The final classifiers are to extract either the relation *SuitsTo* or *SubstitutedBy*. Due to the different nature of the two different relations, the different knowledge sources, classification methods and processing levels may have a different impact on extraction performance. In the following, we discuss these issues in more detail.

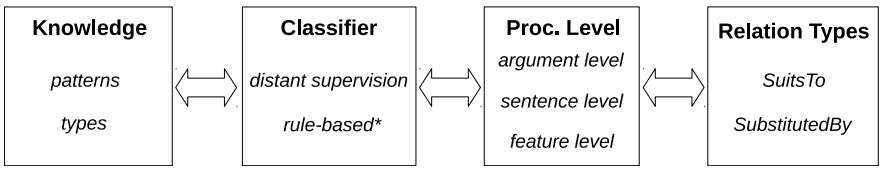

Fig. 1. The most important dependencies examined in this task (*: this classifier cannot be paired with the different *processing levels*).

3.1 Knowledge

Surface Patterns The most straightforward approach to relation extraction is a classification based on surface patterns. We exclusively rely on the set of manually-designed surface patterns introduced in [3] as illustrated in Table 3. A pattern is a lexical sequence comprising the words between two food items.

Table 3. Illustration of the manually designed surface patterns.

Relation	#Patterns	Examples
SuitsTo	8	FOOD and FOOD; FOOD as well as (a) FOOD; FOOD go with FOOD; FOOD with FOOD; FOOD fit to FOOD
SubstitutedBy	8	FOOD or (even) FOOD; FOOD (FOOD; FOOD instead of FOOD; FOOD in place of FOOD; FOOD , resp. FOOD

Type Constraints Another important source of information for relation extraction is intrinsic relation argument information. In the context of the food domain, this corresponds to food type information (since all arguments of the relations we consider are food items). Following [6], we consider the food categorization according to the Food Guide Pyramid [7] (Table 4). In all our experiments, we make use of the best food type model from [6] that had been induced in a semi-supervised manner.

Certain type combinations may be indicative of certain relations. This information could be particularly exploited in situations in which contextual information is inconclusive. For example, Sentences (3) and (4) both contain food

items that match the same surface pattern, i.e. *FOOD and FOOD*. Yet the two sentences convey different relations. A simple type-based rule may help to disambiguate this context (Table 5): If the food items possess different types, then they are more likely to represent instances of *SuitsTo*. However, if the food items are of the same food type, then they are likely to be instances of *SubstitutedBy*.

(3) I very often eat fish$_{MEAT}$ and chips$_{STARCH}$. *(Relation: SuitsTo)*
(4) For these types of dishes you can offer both Burgundy wine$_{BEVER}$ and Champagne$_{BEVER}$. *(Relation: SubstitutedBy)*

Table 4. The different food types.

Category	Description	Size	Perc.
MEAT	meat and fish (products)	394	20.87
BEVER	beverages (incl. alcoholic drinks)	298	15.78
VEGE	vegetables (incl. salads)	231	12.24
SWEET	sweets, pastries and snack mixes	228	12.08
SPICE	spices and sauces	216	11.44
STARCH	starch-based side dishes	185	9.80
MILK	milk products	104	5.51
FRUIT	fruits	94	4.98
GRAIN	grains, nuts and seeds	77	4.08
FAT	fat	41	2.18
EGG	eggs	20	1.06

Table 5. Type assumption for the two target relations.

Relation	Rule
SuitsTo	Is the type pair of the form <x,y>, e.g. <MEAT,STARCH> for <*fish*, *chips*>?
SubstitutedBy	Is the type pair of the form <x,x>, e.g. <FAT,FAT> for <*butter*, *margarine*>?

3.2 Classifiers

Apart from the distantly supervised classifier, we also examine rule-based classification. We designed two different classifiers that exclusively make usage of the two knowledge resources that can also be utilized outside the context of distant supervision, i.e. the manually designed surface patterns and the food type information. The resulting classification algorithms are straightforward. For the ***pattern-based classifier (RB$_{pattern}$)***, we assign to a sentence the label *SuitsTo* or *SubstitutedBy*, in case one of their respective patterns fires. If no pattern fires, then neither of these two relations holds. For the ***type-based classifier (RB$_{type}$)***, we use the type rules from Table 5 to predict either of our two target

relations. Finally, we also include a ***combined classifier (RB$_{comb}$)*** which only predicts a target relation if both the type assumption for the respective relation and one of its patterns fires.

3.3 Processing Levels

In the following, we discuss the three different processing levels we consider for distant supervision. The two knowledge sources, surface patterns and food types, can be applied on all of these levels. We also want to investigate whether the same knowledge resource can have a different impact depending on which level it is applied. Table 6 lists the different methods for the different processing levels.

Table 6. Methods on different processing levels for distant supervision.

Level	Method	Description
argument	random	select argument pairs from relation database at random
	frequency	sort argument pairs according to frequency (of co-occurring in the text corpus)
	pmi	sort argument pairs according to *pointwise mutual information (pmi)*
	patt$^+$	sort argument pairs according to pmi of food items and the surface patterns pertaining to the target relation in *descending* order
	patt$^-$(a)	sort argument pairs according to pmi of food items and the surface patterns pertaining to the target relation in *ascending* order
	patt$^-$(b)	sort argument pairs according to pmi of food items and the surface patterns pertaining to the contrast relation in *descending* order
	type	sort type pairs (e.g. <MEAT,STARCH>) according to pmi and consider their actual food instantiations as arguments
	wup	sort argument pairs according to Wu-&-Palmer [8] similarity in GermaNet [9]
sentence	pattern	only include sentences in which target food items co-occur with surface pattern from target relation
	type	only include sentences in which type rule for the pertaining relation (Table 5) is fulfilled
feature	pattern	include all surface patterns as additional features
	type	include features indicating the types of the target food items, e.g. <MEAT,STARCH> for <*fish*,*chips*>
	standard	standard features directly extracted from training data without external knowledge resources (see Table 7)

Argument-Level Filtering By argument-level filtering, we understand methods by which we **select the arguments** (i.e. entity pairs) from the relation

database to represent typical positive (and negative) relation instances. All these methods have in common that they produce a ranking of argument pairs where the higher ranked pairs are considered more suitable than the lower ranks. For instance, one can sort the argument pairs by their frequency of co-occurrence in our unlabeled text corpus.

We also employ the argument-level filtering for the selection of negative training data, i.e. training instances considered not to convey the target relation. As ***negative*** argument pairs, we consider the union of food pairs of *None* (Table 2) and the pairs of the contrast relation (i.e. *SubstitutedBy* for *SuitsTo* and vice versa[2]). For generating a ranking of argument pairs for negative training data with the help of the surface patterns, we explore two different methods: *patt$^-$(a)* considers the ranking produced by the patterns of the target relation[3], however, the pairs are sorted according to pointwise mutual information in *ascending order*. By that, we mostly aim for food pairs not strongly correlating with the target relation. *patt$^-$(b)*, on the other hand, considers the ranking produced by the patterns of the contrast relation (in *descending order*), so, in this case, we aim for food pairs correlating with the contrast relation.

Another measure that is less self-explanatory is the Wu-&-Palmer [8] similarity computed from GermaNet [9], the German version of WordNet [10] (i.e. *wup*). In principle, this measure indicates the semantic distance of two food items (more precisely, their synsets) in the GermaNet hypernymy graph. It is considered a good measure for detecting (near-)synonyms. We employ it since typical entity pairs for *SubstitutedBy* are similar to (near-)synonyms.[4]

Sentence-Level Filtering By sentence-level filtering, we understand methods by which we **filter the sentences** that match entity pairs from the relation database to represent training data. For example, if we consider a pattern-based sentence filter (i.e. *pattern*), we only include those sentences in which the arguments are connected via a surface pattern of the pertaining target relation (Table 3). For negative training data, we simply *exclude* those sentences that match a particular condition. We apply these filtering methods on the best respective argument-level selection method for the respective relation.

Feature-Level Processing By feature-level processing, we understand the traditional form of feature engineering for supervised learning. Apart from features that make use of either surface patterns or food types, we also include a large

[2] Note that we have two different sets of negative training data depending on what target relation we consider as positive class, i.e. *SuitsTo* or *SubstitutedBy*. We will train two binary classifiers, one for each target relation (versus the remaining instances).

[3] Even though the food pairs do not include instances of the target relation (acc. to the relation database), some of those pairs may still match the patterns of that relation.

[4] Wu-&-Palmer similarity is only used at the argument level. This is due to the fact that the methods at the other processing levels are binary functions rather than continuous functions like Wu-&-Palmer similarity. We did not find an intuitive binary function based on that similarity.

set of standard features that are commonly applied to relation extraction tasks. All these features have in common that they are directly extracted from the sentences which are to be classified. They do not depend on other knowledge resources. The individual standard features are listed in Table 7.

Table 7. Standard features, i.e. features not employing external knowledge.

Feature	**Description**
word-left-window	a window of 2 words to the left of *arg1*
word-right-window	a window of 2 words to the right of *arg2*
word-window	the word sequence between *arg1* and *arg2*
left-lemma-window	a window of 2 words to the left of *arg1* as lemmas
right-lemma-window	a window of 2 words to the right of *arg2* as lemmas
lemma-window	the word sequence between *arg1* and *arg2* as lemmas
bow	all words in the sentence
lemma-bow	lemmas of w_{i-1}, w_{i+1} where $w_i \in \{arg1, arg2\}$
lemma-bigrams	all bigrams $< w_i, w_j >$ between *arg1* and *arg2* as lemmas
pos-left-window	part-of-speech tags of words in a window of 2 words to the left of *arg1*
pos-right-window	part-of-speech tags of words in a window of 2 words to the right of *arg2*
pos-window	part-of-speech sequence between *arg1* and *arg2*
pos-unigrams	part-of-speech tags of w_{i-1}, w_{i+1} where $w_i \in \{arg1, arg2\}$
pos-bigrams	all part-of-speech bigrams $< t_i, t_j >$ between *arg1* and *arg2* using lemmas

3.4 The Two Target Relations

We want to point out that due to the different properties of our two target relations, the demands for a good classifier may vary. By different properties, we specifically mean the notable difference in occurring in our corpus (Table 1). *SubstitutedBy* is a typical *minority* class, while *SuitsTo* is the *majority* class. We assume that for *SuitsTo*, an appropriate classifier needs to focus on precision. With a proportion of 60%, even poor classifiers equivalent to guessing are likely to extract a reasonable amount of true positives, yet precision will be low. With a proportion of only 9%, classifiers for *SubstitutedBy* equivalent to guessing may produce not a single true positive. So, recall also seems important to this relation.

4 Experiments

Even though we explore different methods of producing labeled training data via distant supervision, the training sets always have the same size (10,000 labeled instances). As a supervised classifier, we use Support Vector Machines. As an implementation, we chose SVM^{light}.[5]

[5] `http://svmlight.joachims.org`

For each relation we want to detect (i.e. *SuitsTo* and *SubstitutedBy*), we will build a separate binary classifier, where the positive class is the target relation to be detected, and the negative class are all remaining instances (including instances of the respective contrast relation). We always enforce the class distribution from our gold standard (Table 1). Given a particular ranking of argument pairs for the positive and the negative class, respectively, we randomly sample up to 100 sentences from each pair (where those two food items co-occur)[6], starting from the top of the rankings until the 10,000 instances have been obtained.

4.1 Performance of Argument-Level Filtering

Table 8. F-scores (macro-average) of different argument-level filtering methods (*positive*: filtering methods are applied for the creation of positive training data; *negative*: filtering methods are applied for the creation of negative training data).

	SuitsTo						**SubstitutedBy**					
	positive						***positive***					
negative	**rand.**	**freq**	**pmi**	**wup**	**patt$^+$**	**type**	**rand.**	**freq**	**pmi**	**wup**	**patt$^+$**	**type**
random	41.8	45.0	49.2	46.0	45.3	42.1	61.8	59.0	63.0	59.8	65.1	60.0
freq	40.1	44.1	50.1	41.4	44.8	40.4	61.0	58.1	61.7	59.0	64.0	59.0
pmi	42.3	45.1	50.7	43.5	47.8	43.0	62.8	58.9	64.2	61.0	64.8	59.3
wup	42.4	45.8	50.3	44.4	45.8	42.0	59.3	57.4	60.8	57.6	64.7	55.3
patt$^-$(a)	43.7	47.2	52.2	45.4	47.7	42.0	64.8	62.6	**67.0**	62.9	66.8	63.5
patt$^-$(b)	55.0	56.6	**60.4**	56.7	54.9	54.5	52.5	53.1	57.8	54.7	62.2	50.3
type	42.4	43.8	49.2	44.3	45.9	41.0	61.5	59.3	63.3	59.0	64.6	61.1

We first examine the different argument-level filtering methods. For these experiments, no sentence-level filtering is employed. Moreover, we only train classifiers using only the standard features (Table 7). Table 8 presents the results. All filtering methods are separately applied for the creation of positive training data (all those training data that represent instances of the respective target relation, i.e. *SuitsTo* or *SubstitutedBy*) and negative training data (all those training data that comprise instances not representing instances of the respective target relation). The reason for allowing different filtering methods for those two types of training data is that we must not assume that one single filtering method is optimal for the creation of both positive and negative training data.

The table shows that indeed there is a difference in effectiveness of the methods between the different relations. Moreover, there is also a difference in effectiveness of the methods depending on whether they are applied to rank positive or negative training data. For ranking positive data for *SuitsTo*, *pmi* is the most effective method. For ranking positive data for *SubstitutedBy*, patterns are

[6] In fact, we construct 3 random samples per configuration and report the averaged results. Having 3 individual results per configuration also allows us to apply a paired t-test for statistical significance testing.

slightly more effective than *pmi*. For ranking negative data, for both relations, the best ranking is produced by applying patterns, however, the form of pattern-based ranking is different. That is, for *SubstitutedBy* the best negative ranking comprises arguments that do not correlate with the target relation ($patt^{-}(a)$), while for *SuitsTo* it is those arguments that strongly correlate with the contrast relation ($patt^{-}(b)$). We assume that the reason for this lies in the intrinsic predictiveness of the surface patterns for the different relations (we particularly mean *precision* here, as it is the most relevant evaluation measure to rankings). For *SubstitutedBy*, patterns are more effective than for *SuitsTo* (its precision is 77.72 compared to 55.40 for *SuitsTo* (Table 11:$RB_{pattern}$)). This is why, patterns from *SubstitutedBy* may also serve as negative cues for *SuitsTo*.

In summary, argument-level filtering is very relevant to both relations; for *SuitsTo*, we achieve an increase in F-score by 18.2% points and for *SubstitutedBy* by 5.2% points over the standard random argument selection.

4.2 Performance of Sentence-Level Filtering

Table 9 displays the performance of the different sentence-level filtering methods. We use for each relation the best respective result from argument-level filtering (Table 8). We still use only the standard feature set and apply the filtering methods for the creation of positive and negative training data separately.

The table shows that sentence-level filtering only degrades performance, no matter which combination of methods is chosen. We assume that filtering sentences too drastically reduces the instance diversity for a particular relation. For example, the surface patterns may capture some instances of the pertaining relation, however, if only sentences matching those patterns are included as training data, then many other (representative) instances of that class are excluded. Moreover, the supervised classifier may finally end up learning only the knowledge that is encoded in the surface patterns and nothing beyond it.

Table 9. F-scores (macro-average) of different sentence-level filters (*positive*: filtering methods are applied for the creation of positive training data; *negative*: filtering methods are applied for the creation of negative training data).

SuitsTo				**SubstitutedBy**			
	positive				***positive***		
negative	**no filter**	**pattern**	**type**	***negative***	**no filter**	**pattern**	**type**
no filter	**60.40**	43.22	60.66	**no filter**	**67.00**	64.52	66.05
pattern	47.44	49.19	47.51	**pattern**	28.34	63.66	27.05
type	60.70	43.15	60.90	**type**	67.53	64.50	66.15

4.3 Performance of Feature-Level Processing

Table 10 shows the impact of adding pattern and type information as features. It also displays the impact of argument-level and sentence-level filtering and

contrasts it with the performance of different feature sets. For all classifiers under *feature level*, we use the best respective configuration from argument-level and sentence-level filtering.

The table shows that on feature level, type information is beneficial while patterns are not. This is completely opposite to argument-level filtering. This finding suggests that different types of knowledge sources have varying impact depending on the processing level on which they are applied.

Table 10. F-scores (macro-average) of different processing levels including feature level; *: significantly better than *sentence-level best* at $p < 0.05$ (paired t-test).

	argument level		**sentence level**	**feature level**		
	random	**best**	**best**	**+pattern**	**+type**	**all**
SuitsTo	41.78	60.40	60.90	60.58	61.81*	**61.89***
SubstitutedBy	61.75	67.00	67.53	67.78	70.37*	**70.50***

4.4 Comparing Distant Supervision with Other Classifiers

Table 11 compares distant supervision with a majority classifier (always predicting the majority class), rule-based classification and, as an upper bound, a supervised classifier applying 10-fold cross-validation on our gold standard dataset. Like DS_{best}, the supervised classifier is trained on the best feature set (*all* from Table 10). Even though for distant supervision we increased extraction performance notably by employing argument filters and appropriate feature engineering, for *SuitsTo*, the rule-based classifier exploiting type information produces virtually the same performance as the best distantly supervised classifier.

As already indicated in §3.4, for the relation *SuitsTo* a precision-oriented classifier is most important. Indeed, the best performing classifiers (i.e. DS_{best} and RB_{type}) have a higher precision than the remaining classifiers.

For *SubstitutedBy*, the situation is different. Here, the distantly supervised classifier outperforms all rule-based classifiers. Obviously, for this relation, which represents a minority class, various kinds of information are necessary in order to produce a good classifier. The two basic rule-based classifiers, i.e. $RB_{pattern}$ and RB_{type}, both produce similar F-scores (even though based on different recall/precision levels). From that, we conclude that contextual information *and* types are equally informative. We pointed out in §3.4 that we assume that for *SubstitutedBy*, recall also plays a significant role. Indeed, the classifier with the highest performance, i.e. DS_{best}, exceeds all other classifiers in terms of recall.

In summary, a distantly supervised classifier can produce reasonable performance, however, in some cases, much simpler classifiers, such as RB_{type} for *SuitsTo*, may produce competitive results. Moreover, the best distantly supervised classifier still performs notably worse than a fully supervised upper bound. This suggests that there is still room for improving distantly supervised classification.

Table 11. Comparison of majority classifier, distant supervision, rule-based classifier and supervised classifier (upper bound). We report macro-average precision, recall and F-score.

		Major	**DS**$_{random}$	**DS**$_{best}$	**RB**$_{pattern}$	**RB**$_{type}$	**RB**$_{comb}$	**super**
SuitsTo	**Acc**	60.00	60.84	**68.05**	45.49	68.03	44.73	77.58
	Prec	30.00	53.74	**67.45**	55.40	66.91	59.03	81.32
	Rec	50.00	50.59	62.18	53.13	**62.69**	53.66	72.49
	F	37.50	41.78	61.89	42.60	**62.65**	40.00	73.48
SubstitutedBy	**Acc**	91.00	86.39	86.86	91.47	79.28	**91.56**	92.54
	Prec	45.50	61.28	67.29	77.72	60.55	**81.72**	78.87
	Rec	50.00	62.33	**77.59**	60.28	72.13	57.99	77.31
	F	47.64	61.75	**70.51**	63.97	62.02	61.26	77.77

5 Related Work

The food domain has recently received some attention in the NLP community. Different types of classification have been explored including ontology mapping [11], part-whole relations [12], recipe attributes [13], dish detection and the categorization of food types according to the Food Guide Pyramid [6]. Relation extraction tasks have also been examined. While a strong focus is on food-health relations [14, 15], relations relevant to customer advice have also been addressed [1, 3, 6]. Beyond that, sentiment information was related to food prices with the help of a large corpus consisting of restaurant menus and reviews [16]. Moreover, actionable recipe refinements have been extracted [17]. To the best of our knowledge, we present the first work to investigate the usefulness of distant supervision for relation extraction in the food domain.

6 Conclusion

We examined relation extraction in the food domain by employing distant supervision. We focused on the extraction of two relations that are not only relevant to product recommendation in the food domain, but that also have significance in other domains, such as the fashion or electronics domains. We examined various degrees of freedom in order to select suitable training data. We considered three processing levels being argument level, sentence level and feature level. While argument-level filtering and feature-level processing help to increase classification performance, sentence-level filtering turned out to be not effective. As external resources, we examined manually created surface patterns and semantic types on all these levels. Their effectiveness varies depending on the processing level onto which they are applied. Patterns are effective for argument-level filtering while types can be harnessed on the feature level. We showed that a careful selection of training data and appropriate feature design substantially improves classifi-

cation performance of distantly supervised classifiers, however, in some cases, similar performance can also be achieved by much simpler rule-based methods.

Acknowledgements

Michael Wiegand was funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IC12SO1X. The authors would like to thank Stephanie Köser for annotating the newly created dataset presented in this paper.

References

1. Wiegand, M., Roth, B., Klakow, D.: Data-driven Knowledge Extraction for the Food Domain. In: Proc. of KONVENS. (2012)
2. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant Supervision for Relation Extraction without Labeled Data. In: Proc. of ACL/IJCNLP. (2009)
3. Wiegand, M., Roth, B., Klakow, D.: Web-based Relation Extraction for the Food Domain. In: Proc. of NLDB. (2012)
4. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics **33**(1) (1977) 159–174
5. Wiegand, M., Roth, B., Lasarcyk, E., Köser, S., Klakow, D.: A Gold Standard for Relation Extraction in the Food Domain. In: Proc. of LREC. (2012)
6. Wiegand, M., Roth, B., Klakow, D.: Automatic Food Categorization from Large Unlabeled Corpora and Its Impact on Relation Extraction. In: Proc. of EACL. (2014)
7. U.S. Department of Agriculture, H.N.I.S.: The Food Guide Pyramid. Home and Garden Bulletin 252, Washington, D.C., USA (1992)
8. Wu, Z., Palmer, M.: Verbs semantics and lexcial selection. In: Proc. of ACL. (1994)
9. Hamp, B., Feldweg, H.: GermaNet - a Lexical-Semantic Net for German. In: Proc. of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications. (1997)
10. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-line Lexical Database. International Journal of Lexicography **3** (1990) 235–244
11. van Hage, W.R., Katrenko, S., Schreiber, G.: A Method to Combine Linguistic Ontology-Mapping Techniques. In: Proc. of ISWC. (2005)
12. van Hage, W.R., Kolb, H., Schreiber, G.: A Method for Learning Part-Whole Relations. In: Proc. of ISWC. (2006)
13. Druck, G.: Recipe Attribute Detection Using Review Text as Supervision. In: Proc. of the IJCAI-Workshop on Cooking with Computers. (2013)
14. Miao, Q., Zhang, S., Zhang, B., Meng, Y., Yu, H.: Extracting and Visualizing Semantic Relationships from Chinese Biomedical Text. In: Proc. of PACLIC. (2012)
15. Kang, J.S., Kuznetsova, P., Luca, M., Choi, Y.: Where Not to Eat? Improving Public Policy by Predicting Hygiene Inspections Using Online Reviews. In: Proc. of EMNLP. (2013)
16. Chahuneau, V., Gimpel, K., Routledge, B.R., Scherlis, L., Smith, N.A.: Word Salad: Relating Food Prices and Descriptions. In: Proc. of EMNLP/CoNLL. (2012)
17. Druck, G., Pang, B.: Spice it up? Mining Refinements to Online Instructions from User Generated Content. In: Proc. of ACL. (2012)