ImpAr: A Deterministic Algorithm for Implicit Semantic Role Labelling

Egoitz Laparra
IXA Group
University of the Basque Country
San Sebastian, Spain
egoitz.laparra@ehu.es

German Rigau
IXA Group
University of the Basque Country
San Sebastian, Spain
german.rigau@ehu.es

Abstract

This paper presents a novel deterministic algorithm for implicit Semantic Role Labeling. The system exploits a very simple but relevant discursive property, the argument coherence over different instances of a predicate. The algorithm solves the implicit arguments sequentially, exploiting not only explicit but also the implicit arguments previously solved. In addition, we empirically demonstrate that the algorithm obtains very competitive and robust performances with respect to supervised approaches that require large amounts of costly training data.

1 Introduction

Traditionally, Semantic Role Labeling (SRL) systems have focused in searching the fillers of those explicit roles appearing within sentence boundaries (Gildea and Jurafsky, 2000, 2002; Carreras and Màrquez, 2005; Surdeanu et al., 2008; Hajič et al., 2009). These systems limited their searchspace to the elements that share a syntactical relation with the predicate. However, when the participants of a predicate are implicit this approach obtains incomplete predicative structures with null arguments. The following example includes the gold-standard annotations for a traditional SRL process:

(1) [*arg0* The network] had been expected to have [*np* **losses**] [*arg1* of as much as $20 million] [*arg3* on baseball this year]. It isn't clear how much those [*np* **losses**] may widen because of the short Series.

The previous analysis includes annotations for the nominal predicate **loss** based on the NomBank structure (Meyers et al., 2004). In this case the annotator identifies, in the first sentence, the arguments *arg0*, the entity losing something, *arg1*, the thing lost, and *arg3*, the source of that loss. However, in the second sentence there is another instance of the same predicate, **loss**, but in this case no argument has been associated with it. Traditional SRL systems facing this type of examples are not able to fill the arguments of a predicate because their fillers are not in the same sentence of the predicate. Moreover, these systems also let unfilled arguments occurring in the same sentence, like in the following example:

(2) Quest Medical Inc said it adopted [*arg1* a shareholders' rights] [*np* **plan**] in which rights to purchase shares of common stock will be distributed as a dividend to shareholders of record as of Oct 23.

For the predicate **plan** in the previous sentence, a traditional SRL process only returns the filler for the argument *arg1*, the theme of the plan.

However, in both examples, a reader could easily infer the missing arguments from the surrounding context of the predicate, and determine that in (1) both instances of the predicate share the same arguments and in (2) the missing argument corresponds to the subject of the verb that dominates the predicate, *Quest Medical Inc*. Obviously, this additional annotations could contribute positively to its semantic analysis. In fact, Gerber and Chai (2010) pointed out that implicit arguments can increase the coverage of argument structures in NomBank by 71%. However, current automatic systems require large amounts of manually annotated training data for each predicate. The effort required for this manual annotation explains the absence of generally applicable tools. This problem has become a main concern for many NLP tasks. This fact explains a new trend to develop accurate unsupervised systems that exploit simple but robust linguistic principles (Raghunathan et al., 2010).

In this work, we study the coherence of the predicate and argument realization in discourse. In particular, we have followed a similar approach to

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1180–1189,
Sofia, Bulgaria, August 4-9 2013.

the one proposed by Dahl et al. (1987) who filled the arguments of anaphoric mentions of nominal predicates using previous mentions of the same predicate. We present an extension of this idea assuming that in a coherent document the different ocurrences of a predicate, including both verbal and nominal forms, tend to be mentions of the same event, and thus, they share the same argument fillers. Following this approach, we have developed a deterministic algorithm that obtains competitive results with respect to supervised methods. That is, our system can be applied to any predicate without training data.

The main contributions of this work are the following:

- We empirically prove that there exists a strong discourse relationship between the implicit and explicit argument fillers of the same predicates.
- We propose a deterministic approach that exploits this discoursive property in order to obtain the fillers of implicit arguments.
- We adapt to the implicit SRL problem a classic algorithm for pronoun resolution.
- We develop a robust algorithm, ImpAr, that obtains very competitive results with respect to existing supervised systems. We release an open source prototype implementing this algorithm[1].

The paper is structured as follows. Section 2 discusses the related work. Section 3 presents in detail the data used in our experiments. Section 4 describes our algorithm for implicit argument resolution. Section 5 presents some experiments we have carried out to test the algorithm. Section 6 discusses the results obtained. Finally, section 7 offers some concluding remarks and presents some future research lines.

2 Related Work

The first attempt for the automatic annotation of implicit semantic roles was proposed by Palmer et al. (1986). This work applied selectional restrictions together with coreference chains, in a very specific domain. In a similar approach, Whittemore et al. (1991) also attempted to solve implicit arguments using some manually described semantic constraints for each thematic role they tried to cover. Another early approach was presented by Tetreault (2002). Studying another specific domain, they obtained some probabilistic relations between some roles. These early works agree that the problem is, in fact, a special case of anaphora or coreference resolution.

Recently, the task has been taken up again around two different proposals. On the one hand, Ruppenhofer et al. (2010) presented a task in SemEval-2010 that included an implicit argument identification challenge based on FrameNet (Baker et al., 1998). The corpus for this task consisted in some novel chapters. They covered a wide variety of nominal and verbal predicates, each one having only a small number of instances. Only two systems were presented for this subtask obtaining quite poor results (F1 below 0,02). VENSES++ (Tonelli and Delmonte, 2010) applied a rule based anaphora resolution procedure and semantic similarity between candidates and thematic roles using WordNet (Fellbaum, 1998). The system was tuned in (Tonelli and Delmonte, 2011) improving slightly its performance. SEMAFOR (Chen et al., 2010) is a supervised system that extended an existing semantic role labeler to enlarge the search window to other sentences, replacing the features defined for regular arguments with two new semantic features. Although this system obtained the best performance in the task, data sparseness strongly affected the results. Besides the two systems presented to the task, some other systems have used the same dataset and evaluation metrics. Ruppenhofer et al. (2011), Laparra and Rigau (2012), Gorinski et al. (2013) and Laparra and Rigau (2013) explore alternative linguistic and semantic strategies. These works obtained significant gains over previous approaches. Silberer and Frank (2012) adapted an entity-based coreference resolution model to extend automatically the training corpus. Exploiting this additional data, their system was able to improve previous results. Following this approach Moor et al. (2013) present a corpus of predicate-specific annotations for verbs in the FrameNet paradigm that are aligned with PropBank and VerbNet.

On the other hand, Gerber and Chai (2010, 2012) studied the implicit argument resolution on NomBank. They uses a set of syntactic, semantic and coreferential features to train a logistic regres-

[1] http://adimen.si.ehu.es/web/ImpAr

sion classifier. Unlike the dataset from SemEval-2010 (Ruppenhofer et al., 2010), in this work the authors focused on a small set of ten predicates. But for those predicates, they annotated a large amount of instances in the documents from the Wall Street Journal that were already annotated for PropBank (Palmer et al., 2005) and NomBank. This allowed them to avoid the sparseness problems and generalize properly from the training set. The results of this system were far better than those obtained by the systems that faced the SemEval-2010 dataset. This works represent the deepest study so far of the features that characterizes the implicit arguments [2]. However, many of the most important features are lexically dependent on the predicate and cannot been generalized. Thus, specific annotations are required for each new predicate to be analyzed.

All the works presented in this section agree that implicit arguments must be modeled as a particular case of coreference together with features that include lexical-semantic information, to build selectional preferences. Another common point is the fact that these works try to solve each instance of the implicit arguments independently, without taking into account the previous realizations of the same implicit argument in the document. We propose that these realizations, together with the explicit ones, must maintain a certain coherence along the document and, in consequence, the filler of an argument remains the same along the following instances of that argument until a stronger evidence indicates a change. We also propose that this feature can be exploited independently from the predicate.

3 Datasets

In our experiments, we have focused on the dataset developed in Gerber and Chai (2010, 2012). This dataset (hereinafter BNB which stands for "Beyond NomBank") extends existing predicate annotations for NomBank and ProbBank.

BNB presented the first annotation work of implicit arguments based on PropBank and NomBank frames. This annotation was an extension of the standard training, development and testing sections of Penn TreeBank that have been typically used for SRL evaluation and were already annotated with PropBank and NomBank predicate structures. The authors selected a limited set of predicates. These predicates are all nominalizations of other verbal predicates, without sense ambiguity, that appear frequently in the corpus and tend to have implicit arguments associated with their instances. These constraints allowed them to model enough occurrences of each implicit argument in order to cover adequately all the possible cases appearing in a test document. For each missing argument position they went over all the preceding sentences and annotated all mentions of the filler of that argument. In tables 3 and 4 we show the list of predicates and the resulting figures of this annotation.

In this work we also use the corpus provided for the CoNLL-2008 task. These corpora cover the same BNB documents and include annotated predictions for syntactic dependencies and SuperSense labels as semantic tags. Unlike Gerber and Chai (2010, 2012) we do not use the constituent analysis from the Penn TreeBank.

4 ImpAr algorithm

4.1 Discoursive coherence of predicates

Exploring the training dataset of BNB, we observed a very strong discourse effect on the implicit and explicit argument fillers of the predicates. That is, if several instances of the same predicate appear in a well-written discourse, it is very likely that they maintain the same argument fillers. This property holds when joining the different parts-of-speech of the predicates (nominal or verbal) and the explicit or implicit realizations of the argument fillers. For instance, we observed that 46% of all implicit arguments share the same filler with the previous instance of the same predicate while only 14% of them have a different filler. The remaining 40% of all implicit arguments correspond to first occurrences of their predicates. That is, these fillers can not be recovered from previous instances of their predicates.

The rationale behind this phenomena seems to be simple. When referring to different aspects of the same event, the writer of a coherent document does not repeat redundant information. They refer to previous predicate instances assuming that the reader already recalls the involved participants. That is, the filler of the different instances of a predicate argument maintain a certain discourse coherence. For instance, in example (1), all the argument positions of the second occurrence of the

[2] Gerber and Chai (2012) includes a set of 81 different features.

predicate **loss** are missing, but they can be easily inferred from the previous instance of the same predicate.

(1) [*arg0* The network] had been expected to have [*np* **losses**] [*arg1* of as much as $20 million] [*arg3* on baseball this year]. It isn't clear how much those [*np* **losses**] may widen because of the short Series.

Therefore, we propose to exploit this property in order to capture correctly how the fillers of all predicate arguments evolve through a document.

Our algorithm, **ImpAr**, processes the documents sentence by sentence, assuming that sequences of the same predicate (in its nominal or verbal form) share the same argument fillers (explicit or implicit)[3]. Thus, for every **core** argument arg_n of a predicate, ImpAr stores its previous known filler as a **default** value. If the arguments of a predicate are explicit, they always replace default fillers previously captured. When there is no antecedent for a particular implicit argument arg_n, the algorithm tries to find in the surrounding context which participant is the most likely to be the filler according to some salience factors (see Section 4.2). For the following instances, without an explicit filler for a particular argument position, the algorithm repeats the same selection process and compares the new implicit candidate with the default one. That is, the default implicit argument of a predicate with no antecedent can change every time the algorithm finds a filler with a greater salience. A damping factor is applied to reduce the salience of distant predicates.

4.2 Filling arguments without explicit antecedents

Filling the implicit arguments of a predicate has been identified as a particular case of coreference, very close to pronoun resolution (Silberer and Frank, 2012). Consequently, for those implicit arguments that have not explicit antecedents, we propose an adaptation of a classic algorithm for deterministic pronoun resolution. This component of our algorithm follows the RAP approach (Lappin and Leass, 1994). When our algorithm needs to fill an implicit predicate argument without an explicit antecedent it considers a set of candidates within a window formed by the sentence of the predicate and the two previous sentences. Then, the algorithm performs the following steps:

[3]Note that the algorithm could also consider sequences of closely related predicates.

1. Apply two constraints to the candidate list:
   (a) All candidates that are already explicit arguments of the predicate are ruled out.
   (b) All candidates commanded by the predicate in the dependency tree are ruled out.
2. Select those candidates that are semantically consistent with the semantic category of the implicit argument.
3. Assign a salience score to each candidate.
4. Sort the candidates by their proximity to the predicate of the implicit argument.
5. Select the candidate with the highest salience value.

As a result, the candidate with the highest salience value is selected as the filler of the implicit argument. Thus, this filler with its corresponding salience weight will be also considered in subsequent instances of the same predicate.

Now, we explain each step in more detail using example (2). In this example, arg_0 is missing for the predicate **plan**:

(2) Quest Medical Inc said it adopted [*arg1* a shareholders' rights] [*np* **plan**] in which rights to purchase shares of common stock will be distributed as a dividend to shareholders of record as of Oct 23.

Filtering. In the first step, the algorithm filters out the candidates that are actual explicit arguments of the predicate or have a syntactic dependency with the predicate, and therefore, they are in the search space of a traditional SRL system.

In our example, the filtering process would remove *[a shareholders' rights]* because it is already the explicit argument arg_1, and *[in which rights to purchase shares of common stock will be distributed as a dividend to shareholders of record as of Oct 23]* because it is syntactically commanded by the predicate **plan**.

Semantic consistency. To determine the semantic coherence between the potential candidates and a predicate argument arg_n, we have exploited the selectional preferences in the same way as in previous SRL and implicit argument resolution works. First, we have designed a list of very general semantic categories. Second, we have semi-automatically assigned one of them to every predicate argument arg_n in PropBank and NomBank. For this, we have used the semantic annotation provided by the training documents of the CoNLL-2008 dataset. This annotation was performed automatically using the *SuperSenseTagger* (Ciaramita and Altun, 2006) and includes

named-entities and WordNet Super-Senses[4]. We have also defined a mapping between the semantic classes provided by the SuperSenseTagger and our seven semantic categories (see Table 1 for more details). Then, we have acquired the most common categories of each predicate argument arg_n. ImpAr algorithm also uses the *SuperSenseTagger* over the documents to be processed from BNB to check if the candidate belongs to the expected semantic category of the implicit argument to be filled.

Following the example above, *[Quest Medical Inc]* is tagged as an *ORGANIZATION* by the *SuperSenseTagger*. Therefore, it belongs to our semantic category *COGNITIVE*. As the semantic category for the implicit argument arg_0 for the predicate **plan** has been recognized to be also *COGNITIVE*, *[Quest Medical Inc]* remains in the list of candidates as a possible filler.

Semantic category	Name-entities	Super-Senses
COGNITIVE	*PERSON*	noun.person
	ORGANIZATION	noun.group
	ANIMAL	noun.animal
	...	...
TANGIBLE	*PRODUCT*	noun.artifact
	SUBSTANCE	noun.object
	...	...
EVENTIVE	*GAME*	noun.act
	DISEASE	noun.communication
	...	...
RELATIVE		noun.shape
		noun.attribute
		...
LOCATIVE	*LOCATION*	noun.location
TIME	*DATE*	noun.time
MESURABLE	*QUANTITY*	noun.quantity
	PERCENT	
	...	

Table 1: Links between the semantic categories and some name-entities and super-senses.

Salience weighting. In this process, the algorithm assigns to each candidate a set of salience factors that scores its prominence. The *sentence recency* factor prioritizes the candidates that occur close to the same sentence of the predicate. The *subject*, *direct object*, *indirect object* and *non-adverbial* factors weight the salience of the candidate depending on the syntactic role they belong to. Additionally, the head of these syntactic roles are prioritized by the *head* factor. We have used the same weights, listed in table 2, proposed by Lappin and Leass (1994).

In the example, candidate *[Quest Medical Inc]* is in the same sentence as the predicate **plan**, it belongs to a subject, and, indeed, it is the head of that subject. Hence, the salience score for this candidate is: $100 + 80 + 80 = 260$.

[4]Lexicographic files according to WordNet terminology.

Factor type	weight
Sentence recency	100
Subject	80
Direct object	50
Indirect object	40
Head	80
Non-adverbial	50

Table 2: Weights assigned to each salience factor.

4.3 Damping the salience of the default candidate

As the algorithm maintains the default candidate until an explicit filler appears, potential errors produced in the automatic selection process explained above can spread to distant implicit instances, specially when the salience score of the default candidate is high. In order to reduce the impact of these errors we have included a damping factor that is applied sentence by sentence to the salience value of the default candidate. ImpAr applies that damping factor, r, as follows. It assumes that, independently of the initial salience assigned, 100 points of the salience score came from the *sentence recency* factor. Then, the algorithm changes this value multiplying it by r. So, given a salience score s, the value of the score in a following sentence, s', is:

$$s' = s - 100 + 100 \cdot r$$

Obviously, the value of r must be defined without harming excessively those cases where the default candidate has been correctly identified. For this, we studied in the training dataset the cases of implicit arguments filled with the default candidate. Figure 1 shows that the influence of the default filler is much higher in near sentences that in more distance ones.

We tried to mimic a damping factor following this distribution. That is, to maintain high score salience for the near sentences while strongly decreasing them in the subsequent ones. In this way, if the filler of the implicit argument is wrongly identified, the error only spreads to the nearest instances. If the identification is correct, a lower score for more distance sentences is not too harmful. The distribution shown in figure 1 follows an exponential decay, therefore we have described the damping factor as a curve like the following, where α must be a value within 0 and 1:

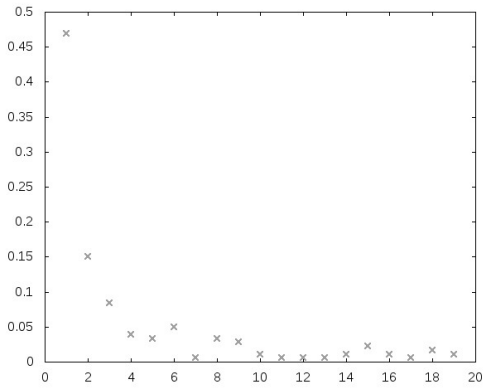

Figure 1: Distances between the implicit argument and the default candidate. The y axis indicate the percentage of cases occurring in each sentence distance, expressed in x

$$r = \alpha^d$$

In this function, d stands for the sentence distance and r for the damping factor to apply in that sentence. In this paper, we have decided to set the value of α to 0.5.

$$r = 0.5^d$$

This value maintains the influence of the default fillers with high salience in near sentences. But it decreases that influence strongly in the following.

In order to illustrate the whole process we will use the previous example. In that case, *[Quest Medical Inc]* is selected as the arg_0 of **plan** with a salience score of 260. Therefore *[Quest Medical Inc]* becomes the default arg_0 of **plan**. In the following sentence the damping factor is:

$$0.5 = 0.5^1$$

Therefore, its salience score changes to $260 - 100 + 100 \cdot 0.5 = 210$. Then, the algorithm changes the default filler for arg_0 only if it finds a candidate that scores higher in their current context. At two sentence distance, the resulting score for the default filler is $260 - 100 + 100 \cdot 0.25 = 185$. In this way, at more distance sentences, the influence of the default filler of arg_0 becomes smaller.

5 Evaluation

In order to evaluate the performance of the ImpAr algorithm, we have followed the evaluation method presented by Gerber and Chai (2010, 2012). For every argument position in the gold-standard the scorer expects a single predicted constituent to fill in. In order to evaluate the correct span of a constituent, a prediction is scored using the Dice coefficient:

$$\frac{2|Predicted \cap True|}{|Predicted| + |True|}$$

The function above relates the set of tokens that form a predicted constituent, $Predicted$, and the set of tokens that are part of an annotated constituent in the gold-standard, $True$. For each missing argument, the gold-standard includes the whole coreference chain of the filler. Therefore, the scorer selects from all coreferent mentions the highest Dice value. If the predicted span does not cover the head of the annotated filler, the scorer returns zero. Then, $Precision$ is calculated by the sum of all prediction scores divided by the number of attempts carried out by the system. $Recall$ is equal to the sum of the prediction scores divided by the number of actual annotations in the gold-standard. F-measure is calculated as the harmonic mean of recall and precision.

Traditionally, there have been two approaches to develop SRL systems, one based on constituent trees and the other one based on syntactic dependencies. Additionally, the evaluation of both types of systems has been performed differently. For constituent based SRL systems the scorers evaluate the correct span of the filler, while for dependency based systems the scorer just check if the systems are able to capture the head token of the filler. As shown above, previous works in implicit argument resolution proposed a metric that involves the correct identification of the whole span of the filler. ImpAr algorithm works with syntactic dependencies and therefore it only returns the head token of the filler. In order to compare our results with previous works, we had to apply some simple heuristics to guess the correct span of the filler. Obviously, this process inserts some noise in the final evaluation.

We have performed a first evaluation over the test set used in (Gerber and Chai, 2010). This dataset contains 437 predicate instances but just 246 argument positions are implicitly filled. Table 3 includes the results obtained by ImpAr, the results of the system presented by Gerber and Chai (2010) and the baseline proposed for the task. Best results are marked in bold[5]. For all predicates, ImpAr improves over the baseline (19.3 points higher in the overall F_1). Our system also outperforms the one presented by Gerber and Chai (2010). Interestingly, both systems present very different performances predicate by predicate. For

[5]No proper significance test can be carried out without the the full predictions of all systems involved.

			Baseline	Gerber & Chai			ImpAr		
	#Inst.	#Imp.	F_1	P	R	F_1	P	R	F_1
sale	64	65	36.2	**47.2**	**41.7**	**44.2**	41.2	39.4	40.3
price	121	53	15.4	36.0	32.6	34.2	**53.3**	**53.3**	**53.3**
investor	78	35	9.8	36.8	**40.0**	38.4	**43.0**	39.5	**41.2**
bid	19	26	32.3	23.8	19.2	21.3	**52.9**	**51.0**	**52.0**
plan	25	20	38.5	**78.6**	**55.0**	**64.7**	40.7	40.7	40.7
cost	25	17	34.8	**61.1**	**64.7**	**62.9**	56.1	50.2	53.0
loss	30	12	52.6	**83.3**	**83.3**	**83.3**	68.4	63.5	65.8
loan	11	9	18.2	**42.9**	**33.3**	**37.5**	25.0	20.0	22.2
investment	21	8	0.0	40.0	25.0	30.8	**47.6**	**35.7**	**40.8**
fund	43	6	0.0	14.3	16.7	15.4	**66.7**	**33.3**	**44.4**
Overall	437	246	26.5	44.5	40.4	42.3	**47.9**	**43.8**	**45.8**

Table 3: Evaluation with the test. The results from (Gerber and Chai, 2010) are included.

			Baseline	Gerber & Chai			ImpAr		
	#Inst.	#Imp.	F_1	P	R	F_1	P	R	F_1
sale	184	181	37.3	**59.2**	**44.8**	**51.0**	44.3	43.3	43.8
price	216	138	34.6	**56.0**	48.7	52.1	55.0	**54.5**	**54.7**
investor	160	108	5.1	**46.7**	39.8	**43.0**	28.2	27.0	27.6
bid	88	124	23.8	60.0	36.3	**45.2**	48.4	**41.8**	45.0
plan	100	77	32.3	**59.6**	44.1	**50.7**	47.0	**47.0**	47.0
cost	101	86	17.8	**62.5**	**50.9**	**56.1**	49.2	43.7	46.2
loss	104	62	54.7	**72.5**	**59.7**	**65.5**	63.0	58.2	60.5
loan	84	82	31.2	**67.2**	**50.0**	**57.3**	56.4	45.6	50.6
investment	102	52	15.5	32.9	**34.2**	33.6	**41.2**	30.9	**35.4**
fund	108	56	15.5	**80.0**	35.7	49.4	55.6	**44.6**	**49.5**
Overall	1,247	966	28.9	**57.9**	**44.5**	**50.3**	47.7	43.0	45.3

Table 4: Evaluation with the full dataset. The results from (Gerber and Chai, 2012) are included.

instance, our system obtains much higher results for the predicates **bid** and **fund**, while much lower for **loss** and **loan**. In general, ImpAr seems to be more robust since it obtains similar performances for all predicates. In fact, the standard deviation, σ , of F_1 measure is 10.98 for ImpAr while this value for the (Gerber and Chai, 2010) system is 20.00.

In a more recent work, Gerber and Chai (2012) presented some improvements of their previous results. In this work, they extended the evaluation of their model using the whole dataset and not just the testing documents. Applying a cross-validated approach they tried to solve some problems that they found in the previous evaluation, like the small size of the testing set. For this work, they also studied a wider set of features, specially, they experimented with some statistics learnt from parts of GigaWord automatically annotated. Table 4 shows that the improvement over their previous system was remarkable. The system also seems to be more stable across predicates. For comparison purposes, we also included the performance of ImpAr applied over the whole dataset.

The results in table 4 show that, although ImpAr still achieves the best results in some cases, this time, it cannot beat the overall results obtained by the supervised model. In fact, both systems obtain a very similar recall, but the system from (Gerber and Chai, 2012) obtains much higher precision. In both cases, the σ value of F_1 is reduced, 8.81 for ImpAr and 8.21 for (Gerber and Chai, 2012). However, ImpAr obtains very similar performance independently of the testing dataset what proves the robustness of the algorithm. This suggests that our algorithm can obtain strong results also for other corpus and predicates. Instead, the supervised approach would need a large amount of manual annotations for every predicate to be processed.

6 Discussion

6.1 Component Analysis

In order to assess the contribution of each system component, we also tested the performance of ImpAr algorithm when disabling only one of its components. With this evaluations we pretend to sight the particular contribution of each component. In table 5 we present the results obtained in the following experiments for the two testing sets explained in section 5:

- Exp1: The damping factor is disabled. All selected fillers maintain the same salience over

all sentences.

- Exp2: Only explicit fillers are considered as candidates[6].
- Exp3: No default fillers are considered as candidates.

As expected, we observe a very similar performances in both datasets. Additionally, the highest loss appears when the default fillers are ruled out (Exp3). In particular, it also seems that the explicit information from previous predicates provides the most correct evidence (Exp2). Also note that for Exp2, the system obtains the highest precision. This means that the most accurate cases are obtained by previous explicit antecedents.

	test			full		
	P	R	F_1	P	R	F_1
full	47.9	**43.8**	**45.8**	47.7	**43.0**	**45.3**
Exp1	45.7	41.8	43.6	47.1	42.5	44.8
Exp2	**51.2**	*24.6*	33.2	**55.3**	*25.5*	34.9
Exp3	*34.6*	29.7	*31.9*	*34.8*	28.9	*31.5*
Exp4	**42.6**	**37.9**	**40.1**	**37.5**	**31.2**	**34.1**
Exp5	*38.8*	*34.5*	*36.5*	*35.7*	*29.7*	*32.4*
Exp6	53.3	48.7	50.9	52.4	47.2	49.6

Table 5: Exp1, Exp2 and Exp3 correspond to ablations of the components. Exp3 and Exp4 are experiments over the cases that are not solved by explicit antecedents. Exp6 evaluates the system capturing just the head tokens of the constituents.

As Exp1 also includes instances with explicit antecedents, and for these cases the damping factor component has no effect, we have designed two additional experiments:

- Exp4: Full system for the cases not solved by explicit antecedents.
- Exp5: As in Exp4 but with the damping factor disabled.

As expected, now the contribution of the dumping factor seems to be more relevant, in particular, for the *test* dataset.

6.2 Correct span of the fillers

As explained in Section 5, our algorithm works with syntactic dependencies and its predictions only return the head token of the filler. Obtaining the correct constituents from syntactic dependencies is not trivial. In this work we have applied a simple heuristic that returns all the descendant tokens of the predicted head token. This naive process inserts some noise to the evaluation of the system. For example, from the following sentence our system gives the following prediction for an implicit arg_1 of an instance of the predicate **sale**:

> Ports of Call Inc. reached agreements to sell its remaining seven aircraft [arg_1 to buyers] that weren't disclosed.

But the actual gold-standard annotation is: [arg_1 buyers that weren't disclosed]. Although the head of the constituent, *buyers*, is correctly captured by ImpAr, the final prediction is heavily penalized by the scoring method. Table 5 presents the results of ImpAr when evaluating the head tokens of the constituents only (Exp6). These results show that the current performance of our system can be easily improved applying a more accurate process for capturing the correct span.

7 Conclusions and Future Work

In this work we have presented a robust deterministic approach for implicit Semantic Role Labeling. The method exploits a very simple but relevant discoursive coherence property that holds over explicit and implicit arguments of closely related nominal and verbal predicates. This property states that if several instances of the same predicate appear in a well-written discourse, it is very likely that they maintain the same argument fillers. We have shown the importance of this phenomenon for recovering the implicit information about semantic roles. To our knowledge, this is the first empirical study that proves this phenomenon.

Based on these observations, we have developed a new deterministic algorithm, ImpAr, that obtains very competitive and robust performances with respect to supervised approaches. That is, it can be applied where there is no available manual annotations to train. The code of this algorithm is publicly available and can be applied to any document. As input it only needs the document with explicit semantic role labeling and Super-Sense annotations. These annotations can be easily obtained from plain text using available tools[7], what makes this algorithm the first effective tool available for implicit SRL.

As it can be easily seen, ImpAr has a large margin for improvement. For instance, providing more accurate spans for the fillers. We also plan

[6] That is, implicit arguments without explicit antecedents are not filled.

[7] We recommend mate-tools (Björkelund et al., 2009) and SuperSenseTagger (Ciaramita and Altun, 2006).

to test alternative approaches to solve the arguments without explicit antecedents. For instance, our system can also profit from additional annotations like coreference, that has proved its utility in previous works. Finally, we also plan to study our approach on different languages and datasets (for instance, the SemEval-2010 dataset).

8 Acknowledgment

We are grateful to the anonymous reviewers for their insightful comments. This work has been partially funded by SKaTer (TIN2012-38584-C06-02), OpeNER (FP7-ICT-2011-SME-DCL-296451) and NewsReader (FP7-ICT-2011-8-316404), as well as the READERS project with the financial support of MINECO, ANR (convention ANR-12-CHRI-0004-03) and EPSRC (EP/K017845/1) in the framework of ERA-NET CHIST-ERA (UE FP7/2007-2013).

References

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, ACL '98, Montreal, Quebec, Canada, pp. 86–90.

Björkelund, A., L. Hafdell, and P. Nugues (2009). Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, Boulder, Colorado, USA, pp. 43–48.

Carreras, X. and L. Màrquez (2005). Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, CoNLL '05, Ann Arbor, Michigan, USA, pp. 152–164.

Chen, D., N. Schneider, D. Das, and N. A. Smith (2010). Semafor: Frame argument resolution with log-linear models. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, Los Angeles, California, USA, pp. 264–267.

Ciaramita, M. and Y. Altun (2006). Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, Sydney, Australia, pp. 594–602.

Dahl, D. A., M. S. Palmer, and R. J. Passonneau (1987). Nominalizations in pundit. In *In Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, ACL '87, Stanford, California, USA, pp. 131–139.

Fellbaum, C. (1998). *WordNet: an electronic lexical database*. MIT Press.

Gerber, M. and J. Chai (2012, December). Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics 38*(4), 755–798.

Gerber, M. and J. Y. Chai (2010). Beyond nombank: a study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, Uppsala, Sweden, pp. 1583–1592.

Gildea, D. and D. Jurafsky (2000). Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, Hong Kong, pp. 512–520.

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

Gorinski, P., J. Ruppenhofer, and C. Sporleder (2013). Towards weakly supervised resolution of null instantiations. In *Proceedings of the 10th International Conference on Computational Semantics*, IWCS '13, Potsdam, Germany, pp. 119–130.

Hajič, J., M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang (2009). The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, Boulder, Colorado, USA, pp. 1–18.

Laparra, E. and G. Rigau (2012). Exploiting explicit annotations and semantic types for implicit argument resolution. In *6th IEEE International Conference on Semantic Computing*, ICSC '12, Palermo, Italy, pp. 75–78.

Laparra, E. and G. Rigau (2013). Sources of evidence for implicit argument resolution. In *Proceedings of the 10th International Conference on Computational Semantics*, IWCS '13, Potsdam, Germany, pp. 155–166.

Lappin, S. and H. J. Leass (1994, December). An algorithm for pronominal anaphora resolution. *Computational Linguistics 20*(4), 535–561.

Meyers, A., R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman (2004). The nombank project: An interim report. In *In Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation*, HLT-NAACL '04, Boston, Massachusetts, USA, pp. 24–31.

Moor, T., M. Roth, and A. Frank (2013). Predicate-specific annotations for implicit role binding: Corpus annotation, data analysis and evaluation experiments. In *Proceedings of the 10th International Conference on Computational Semantics*, IWCS '13, Potsdam, Germany, pp. 369–375.

Palmer, M., D. Gildea, and P. Kingsbury (2005, March). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics 31*(1), 71–106.

Palmer, M. S., D. A. Dahl, R. J. Schiffman, L. Hirschman, M. Linebarger, and J. Dowding (1986). Recovering implicit information. In *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, ACL '86, New York, New York, USA, pp. 10–19.

Raghunathan, K., H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning (2010). A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, Cambridge, Massachusetts, USA, pp. 492–501.

Ruppenhofer, J., P. Gorinski, and C. Sporleder (2011). In search of missing arguments: A linguistic approach. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, RANLP '11, Hissar, Bulgaria, pp. 331–338.

Ruppenhofer, J., C. Sporleder, R. Morante, C. Baker, and M. Palmer (2010). Semeval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, Los Angeles, California, USA, pp. 45–50.

Silberer, C. and A. Frank (2012). Casting implicit role linking as an anaphora resolution task. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, *SEM '12, Montréal, Canada, pp. 1–10.

Surdeanu, M., R. Johansson, A. Meyers, L. Màrquez, and J. Nivre (2008). The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Natural Language Learning*, CoNLL '08, Manchester, United Kingdom, pp. 159–177.

Tetreault, J. R. (2002). Implicit role reference. In *International Symposium on Reference Resolution for Natural Language Processing*, Alicante, Spain, pp. 109–115.

Tonelli, S. and R. Delmonte (2010). Venses++: Adapting a deep semantic processing system to the identification of null instantiations. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, Los Angeles, California, USA, pp. 296–299.

Tonelli, S. and R. Delmonte (2011). Desperately seeking implicit arguments in text. In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*, RELMS '11, Portland, Oregon, USA, pp. 54–62.

Whittemore, G., M. Macpherson, and G. Carlson (1991). Event-building through role-filling and anaphora resolution. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, ACL '91, Berkeley, California, USA, pp. 17–24.