UNIVERSIDADE FEDERAL FLUMINENSE

**BRENO WILLIAM SANTOS REZENDE DE CARVALHO**

# AUGMENTING LINGUISTIC SEMI-STRUCTURED DATA FOR MACHINE LEARNING

NITERÓI

2018

UNIVERSIDADE FEDERAL FLUMINENSE

**BRENO WILLIAM SANTOS REZENDE DE CARVALHO**

# AUGMENTING LINGUISTIC SEMI-STRUCTURED DATA FOR MACHINE LEARNING

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação

Orientadora:
ALINE MARINS PAES CARVALHO

Co-orientador:
BERNARDO GONÇALVES

NITERÓI

2018

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

C331a Carvalho, Breno William Santos Rezende de
Augmenting Linguistic Semi-Structured Data for Machine Learning : / Breno William Santos Rezende de Carvalho ; Aline Marins Paes Carvalho, orientador ; Bernardo Nunes Gonçalves, coorientador. Niterói, 2018.
57 p. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2018.

DOI: http://dx.doi.org/10.22409/PGC.2018.m.12883015708

1. Aprendizado de máquina. 2. Automatic Knowledge Base Construction. 3. Semântica. 4. Processamento de linguagem natural (Computação) . 5. Produção intelectual. I. Carvalho, Aline Marins Paes, orientador. II. Gonçalves, Bernardo Nunes, coorientador. III. Universidade Federal Fluminense. Escola de Engenharia. IV. Título.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

BRENO WILLIAM SANTOS REZENDE DE CARVALHO

# AUGMENTING LINGUISTIC SEMI-STRUCTURED DATA FOR MACHINE LEARNING

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação

Aprovada em Dezembro de 2018.

BANCA EXAMINADORA

D.Sc. ALINE MARINS PAES CARVALHO - Orientadora, PCG/UFF

D.Sc. BERNARDO GONÇALVES - Co-orientador, IBM Research

Ph.D. BIANCA ZADROZNY, IBM Research

D.Sc. FLAVIA CRISTINA BERNARDINI, PCG/UFF

Ph.D. FABIO GAGLIARDI COZMAN, UFRJ

Niterói

2018

*I dedicate this work to everyone and each one who have ever used their curiosity and perseverance to make this world a better place.*

# Acknowledgements

Fundamental to the conclusion of this dissertation were the teachings and faith in this project of my advisors, Aline and Bernardo. I am deeply grateful for their guidance and commitment. Aline, as my teacher in the Artificial Intelligence course and my advisor during my graduation — and now also as my advisor— you shaped my view of the field and of research in a broader sense. I can genuinely say that I am a better person because I knew you. The impact you had in my life is precious to me, and I will never be able to thank you enough. Bernardo, thank you so much for all the teaching, all the conversations, and for your valuable support. Thank you both for the knowledge, the good humor, the honest responses and for being there every time I needed.

I am also grateful to the dissertation committee, composed by Bianca Zadrozny, prof. Fabio Cozman and Flavia Bernardini, your insightful comments were precious. I also want to thank prof. Bruno Lopes for gracefully accepting the position of substitution member.

To my family, I want to express my most profound appreciation, for giving me support during this project, and also during every other step I took, in my life. To my parents, thank you for your ever-present love and reassuring words. Karen and Kátia, my sisters, thank you for always being there for me.

Words are such magical things, they can bring light, strength and calm in such serendipitous ways. I am especially grateful for the needed sheltering and encouraging words of many of my friends. Patricia Carrion, thank you for all the advice, tips, friendship and support. Leandro, thank you for your patience and love. Daniela Szwarcman, for the fruitful conversations and for being an example of dedication and personal interest in the work, you have my gratitude. Bianca Zadrozny, once again thank you for all the support and inspiration, you are a great leader. Guilherme Paulino Passos, Luan Teylo, and Rodrigo Alves, I want to thank you for the best conversations about big and important questions, and also for also the long and meaningful discussions about utterly useless things. André thank you for all the advice. Rafael Drummond for the perspective and for the mornings running together.

# Resumo

A tarefa de Semantic Role Labeling (SRL) consiste em encontrar os papéis semânticos dos termos em uma frase de maneira automática. Esta é uma tarefa essencial para a criação de uma representação estruturada automática do significado de um fragmento de texto. Um recurso linguístico conhecido na literatura para essa tarefa é o conjunto de documentos manualmente anotados disponíveis no Projeto FrameNet. A FrameNet é um banco de dados léxico legível por homens e máquinas, contendo um número considerável de sentenças anotadas que compõem a estrutura de *frames*. No entanto, enquanto vários *frames* são fornecidos com sentenças anotadas, um grande grupo deles não possui anotações úteis. Neste trabalho, apresentamos um método de aumento de dados para os documentos da FrameNet. Esta técnica aumenta em mais de 13 % o número total de anotações. Para tanto, nos apoiamos em aspectos lexicos, sintáticos e semânticos das sentenças. Avaliamos o método de aumento proposto comparando o desempenho de um método de *semantic role labeling* de última geração, com e sem aumento.

**Palavras-chave**: FrameNet, Frame Semantic Parsing, Semantic Role Labeling, Aumento de Dados.

# Abstract

Semantic Role Labeling (SRL) is the process of automatically finding the semantic roles of terms in a sentence. It is an essential task towards creating a machine-meaningful representation of textual information. One well-known supportive resource for this task is the set of manually annotated documents available in the FrameNet Project. FrameNet is a human and machine-readable lexical database containing a considerable number of annotated sentences that compose the frames structure. However, while a number of frames are provided with annotated sentences, a large group of them lacks useful annotations. In this work, we present a data augmentation method for FrameNet documents that increases by over 13% the total number of annotations. It relies on lexical, syntactic and semantic aspects of the sentences. We evaluate the proposed augmentation method by comparing the performance of a state-of-the-art semantic-role-labeling method, with and without augmentation.

**Keywords**: Automatic Knowledge Base Construction, Frame Semantic Parsing, Frames, FrameNet, SemEval'07 task 19, Semantic Role Labeling, Statistical Relational Artificial Intelligence, StarAI, Semantic Parsing.

# List of Figures

# List of Tables

# List of Abbreviations and Acronyms

| | | |
|---|---|---|
| API | : | Application Program Interface; |
| CCG | : | Combinatory Categorial Grammar; |
| DRS | : | Discourse Representation Structure; |
| DRT | : | Discourse Representation Theory; |
| FSP | : | Frame Semantic Parsing; |
| LF | : | Logical Form; |
| NLP | : | Natural Language Processing; |
| PoS | : | Part-of-Speech; |
| Seg-RNN | : | Segmental Recurrent Neural Network; |
| SRL | : | Semantic Role Labeling; |
| StarAI | : | Statistical Relational Artificial Intelligence |

# Contents

# Chapter 1

# Introduction

A large portion of humankind knowledge is (still) stored in written form. For instance, Wikipedia, the largest used and well-known free and collaborative encyclopedia, had over five million articles and almost half hundred million pages about nearly any subject ever considered by the time this dissertation was completed [50]. Nevertheless, much of this information is unstructured. Unstructured information, in this scenario, means that they are not represented in a data structure that favors algorithmic processing. Arguably, such data structure should make possible to machines efficiently manipulate the semantics of a sentence.

Not having an underlying data structure representing the meaning of the components of the sentence and their relationships makes it more difficult to search, catalog and query a text corpora, as mentioned in [5]. Any query more complicated than a keyword search can benefit from a semantic representation of the textual information available nowadays. For instance, semantic representation empowers reasoning systems to respond queries like "How much time would a spaceship take to cross the Milkway, departing from Earth?", even if this information is not explicit in the text corpora. Considering that there is a semantic representation of the facts (i) "Light would take at least a hundred thousand years to cross the Milkway, departing from Earth", (ii) "No object travels faster than light" and (iii) "Spaceships are objects"; then, the system would be able to answer "The spaceship would take at least a thousand hundred years."

However, unstructured textual data would require a monumental effort to be annotated by humans into semantic representations that are machine-friendly, see [47] a system intended to mitigate some of this effort. This limitation creates the need for automated extraction of information, making the text amenable for automatic querying of the underlying meaning of the textual information. One of the most straightforward methods of

semantic annotations of sentences is to annotate the semantic role of the entities present in the text. In the sentence "Mary went to the store to buy an ice cream", Mary is a buyer, ice cream is a product, and store is a place, and all of those entities and their roles are related to each other through the concept **Buy**. Such kind of annotation is called Semantic Role Labeling.

Semantic Role Labeling, or SRL for short [1], is a task that relates to the emerging area of Machine Reading [15], a sub-area within Natural Language Processing (NLP). Machine Reading is concerned with creating machine friendly representation of the meaning of a given piece of text, while NLP is concerned with the more general task of processing any text document, it can be syntactical, semantic, pragmatic, etc. SRL is specifically concerned with creating machine-friendly, yet nuanced, representations of text, and it consists of mapping elements of a given sentence to predefined sets of *semantic roles*.

Semantic roles, in turn, are roles that can be attributed to sentences in a specific semantic context. There are two main kinds of semantic role labeling: *deep* and *shallow*. The *deep* labeling maps tokens of the sentence to somewhat complex semantic structures, a step beside the scope of this dissertation. The *shallow* labeling, in turn, consists in mapping the tokens to an abstract semantic role. For instance, figure 1.1 shows two shallow roles, namely, Content and Paradigm, which provide meaning to two subsets of tokens in the sentence.

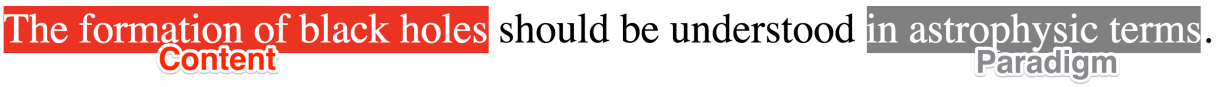



Figure 1.1: An example of shallow semantic roles assigned to tokens in a sentence.

In this dissertation, we are concerned with shallow labeling, which is itself far from a trivial computational task. There is a rich literature on automatic SRL parsers, e.g., [10, 41, 48, 51], the most recent ones relying on statistical methods, in specific, Machine Learning methods, for training. Supervised statistical methods require a large and useful set of labeled examples, in this case, annotated sentences, whereby "good" we mean a set of sentences whose tokens are annotated with their expected deep roles. It is also essential that those annotations cover a large number of all the existent (already specified) semantic roles, that would be said to be a good coverage.

There is a number of sources of annotated sentences to support Machine Reading, like PropBank [28], VerbNet [29], FrameNet [16], and so on. Each of them has been used for specific tasks in Machine Reading. Linguistic resources like those are highly valuable

to the NLP community since they not only provide labeled data for supervised statistical methods, but also provide quantitative benchmarks for comparing models. Since FrameNet is an important resource used for SRL,[31], any improvement in its coverage and quantity of examples can directly impact many statistical models that rely on this data.

FrameNet is a publicly-available linguistic resource, and it consists in a network of concepts (so-called frames) such as **Run**, **Motive** and **Location**. Each frame is composed of frame elements, which define semantic roles in the domains[1]. A technical challenge, however, is that FrameNet's set distribution of annotated sentences forms a long tail — only a few frame elements have several examples, while most of them have only one or none example at all —, making it difficult to tackle less popular frame elements. This matter gets even more pressing when we target specific domains within FrameNet.

This dissertation main research question consists on whether a data augmentation method that enlarges the set of annotations and its distribution in FrameNet, would improve the performance of automatic Semantic Role Labelers or not. In this regard, we carried out matching of frame elements over different frames — under either notion of lexical, syntactic or semantic equivalence — so that sentences received new (inferred) annotations. We took advantage of the inter-frame connections to enrich the information available in the resource. In addition to our primary aim, we provided to the community an alternative Python API for manipulation of the FrameNet components. The API also allows the manipulation of the annotated documents.

The rest of this dissertation is organized as follows. The chapter 2 presents an overview preparing towards our research problem. We start with a brief introduction to NLP and how SRL fits in this area. Doing so, we describe the research materials used, the semantic and syntactic analyzers that will enable us to parse natural language sentences and feed our augmentation method. We also provide and a more in-depth view on FrameNet, the linguistic resource that contains the corpora of annotated documents that can be used for SRL. We also described the aspects of the frame relationships and structure that are relevant to this dissertation.

Chapter 3 contains the description of our augmentation method, the main contribution of this dissertation. After a short explanation of the importance of data augmentation for SRL parsers, we provide an overview of the method and then dig into its details through

[1]There is the general task of labeling sentences (SRL) in the literature, while in the FrameNet's jargon it is usually called argument identification [12].

a practical example. In chapter 4, we detail our experimental methodology for evaluating our augmentation method. We explore different strategies and settings and discuss how they impacted the overall performance of our method. We also provide a description of the data used and their original coverage. There is a discussion, at the end of this chapter, about the results observed.

In chapter 5 we situate this work within the literature through a discussion of related work. In chapter 6 we conclude the paper, and point challenges and future work.

# Chapter 2

# Background

Natural Language Processing, or NLP for short, is the research area in Computer Science concerned with creating a formal representation of textual information[8]. Those representations can be lexical, syntactic, semantic, pragmatic or discourse-oriented. Each one of those representations focuses on a different aspect of the underlying information contained in a piece of text.

Lexical parsers are the most straightforward parsers. They usually are concerned with two specific tasks, tokenization, and lemmatization. Tokenization is the task of splitting a given sentence in its correct tokens, often words and punctuation. Lemmatization in the other hand is the task of giving a word, find its lemma, i.e., the canonical form of the word. For example, the lemma of the word "running" is "run". For an example of the use of such parsers, we refer the user to the book [8].

Syntactic parsers are concerned with the structure of the sentence and how each token relates to each other. There are mainly two kinds of syntactic representations, the dependency tree, and the constituency tree. The dependency tree of a sentence is built from the major verb of the sentence and then linking each token that immediately relates to this verb. This process is repeated until all tokens are connected in the tree. This way, each node in this tree is a token. The constituency tree, on the other hand, sees the sentence as being composed by other sub-sentences that are related to each other. This way, the root node of a sentence constituency tree represents the sentence itself. Each child node represents a sub-sentence. Continuing this process to each sub-sentences, one would end up on the tokens of the original sentence. Thus, only the leaf nodes of this kind of tree are tokens from the sentence.

Semantic analyzers aim at creating a structured representation of the underlying

meaning of a given sentence, or a piece of text. They usually rely on syntactic parsers for generating a preliminary representation from where the semantic representation is built upon. A number of representations have been proposed in the literature and one common goal for such representations seem to be shared by the community. This common goal is that those representations should be syntactically robust. I.e., if two sentences state the same semantic information and are just different ways of stating it, then they should have the same semantic representation. In achieving so, a method usually needs to solve the *anaphora resolution problem*. This problem consists of matching the pronouns that appear in the text to the correct entities that are stated somewhere else in the text.

Another frequent problem faced by the literature is the passive voice occurrence. For instance, the sentences "We observed the Moon" and "The moon was observed by us" carry the same underlying fact, but might be represented in different ways, the first would be "observe(we, moon)" and the second one "be_observed(moon, we)".

Many different semantic representations have been proposed. Some of them consist of the labeling of entities in the sentences and their relationships. Some of them use a rather limited number of labels, usually thematic roles, while other uses a wide pool of semantic roles extracted from a linguistic resource. This representation is going to be further explored in section 2.3.

In the next sections of this chapter, we present the syntactic and semantic parsers used in this dissertation; spaCy dependency tree and Boxer, respectively. We also provide a description of the linguistic resource FrameNet since it provides a training dataset and benchmark for SRL and describes the SRL task, the task chosen to evaluate our augmentation method. We conclude by outlining Open-Sesame, the semantic-role-labeling method that supports our evaluation method (section 2.4)

## 2.1 Sentence Analyzers: syntax and semantics

Boxer and spaCy are, respectively, the semantic and syntactic parsers used in this dissertation and are described in the next sections (sections 2.1.1 and 2.1.2). They parse the sentence into different representations that we convert into a common Logical Form to translate them into an uniform representation. This logical form is a uniform way of representing syntax and semantics and is used in all the steps of our augmentation method.

### 2.1.1 SpaCy Dependency Tree

As our syntactic analyzer, we use the dependency tree parser and part-of-speech tagging system provided by the spaCy NLP library [46] (version 2.0.11).

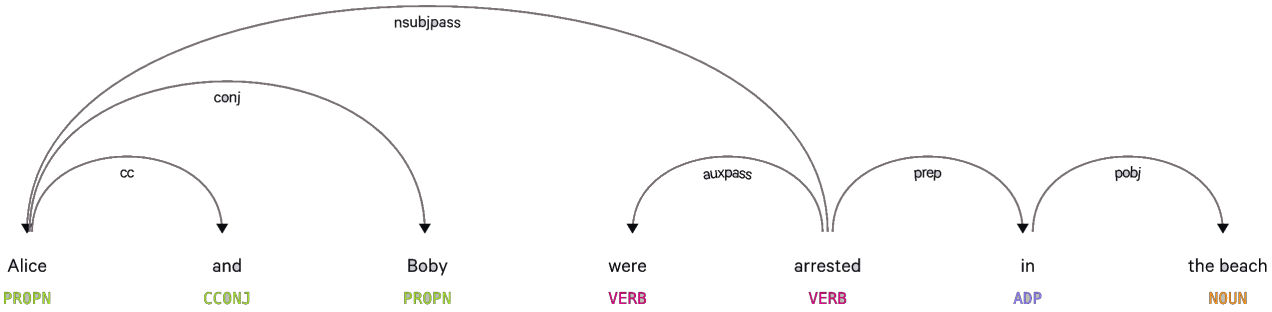



Figure 2.1: Syntactic Analysis by spaCy

Part-of-speech tagging consists of labeling tokens in the sentence (not only words but also punctuation) with their part-of-speech, such as verbs, nouns, and pronouns adjectives. The dependency tree is a syntactic representation of the sentence that explicit the relationships among the sentence tokens, it states the relations of 'head' tokens and tokens that modify them. It starts from the main verb of the sentence as the root of the tree and then traverses through the immediately related tokens that modify this verb.

Figure2.1 shows spaCy in action. The node labels (associated with the sentence tokens, e.g., 'VERB') give the part-of-speech tags, and the edge labels (associated with the tokens relationships, e.g., 'conj') are dependency tags.

### 2.1.2 Boxer

Boxer is an open-domain semantic analyzer [9], based on Combinatorial Categorical Grammars (CCG)[4], and Discourse Representation Theory (DRT)[14]. It generates a neo-Davidsonian representation of sentences.

#### 2.1.2.1 Combinatorial Categorical Grammars

CCG is an efficient grammar formalism that is fast to parse. It is based on constituency-structures, opposed to dependency structures like the one we use from spaCy, and it has a straightforward relationship between syntax and semantics. This formalism is a fusion of Combinatory Logics, a notation to avoid quantifier in mathematical logic, and Categorical

Grammar, a family of formalisms motivated by the principle of compositionality of the syntactic components of a sentence. This being said, in the CCG formalism, syntactic elements are mapped to categories that associate them to functions and specifies the type and directionality of their arguments and the type of the result of those functions.

CCG formalisms have the concept of combinators, and those combinators provide a way to escape the problem of coordinating contiguous strings that do not embody constituents but are still related. A detailed description of this formalism is found in [4]. Each token in a sentence has a syntactic and semantic component derived from the lexicon of the grammar. The syntactic component is a composition of PoS tags while the semantics is usually an expression represented in lambda calculus that is used to compose the meaning of the entire sentence.

The syntactic composition is achieved by the operators "\" and "/" that state if the term can be composed with the terms in its right or left.

Figure 2.2a shows the sentence "CCG is fun" with the Part-of-Speech tagging of each token and the semantics expression associated with this lexicon. After that, we see the application of a backward reduction in Figure 2.2b. Figure 2.2b shows the application of a forward reduction that finishes the parsing of the sentence.

| CCG | is | fun |
|---|---|---|
| $NP$ : $CCG$ | $S\backslash NP\,/ADJ$ : $\lambda f.\lambda x.f(x)$ | $ADJ$ : $\lambda x.fun(x)$ |

(a) The sentence, the syntactic tags and the semantics

| CCG | is | fun |
|---|---|---|
| $NP$ : $CCG$ | $S\backslash NP\,/ADJ$ : $\lambda f.\lambda x.f(x)$ | $ADJ$ : $\lambda x.fun(x)$ |
| | $S\backslash NP : \lambda x.fun(x)$ (<) | |

(b) Backward reduction of "is" and "fun"

| CCG | is | fun |
|---|---|---|
| $NP$ : $CCG$ | $S\backslash NP\,/ADJ$ : $\lambda f.\lambda x.f(x)$ | $ADJ$ : $\lambda x.fun(x)$ |
| | $S\backslash NP : \lambda x.fun(x)$ (<) | |
| $S : fun(CCG)$ (>) | | |

(c) Forward reduction

Figure 2.2: Example of CCG parsing

#### 2.1.2.2 Discourse Representation Theory

DRT is a formalism introduced by [24]. It interprets each sentence in terms of its contribution to an existing piece of already interpreted discourse. Thus each new element updates the representation of the discourse [14].

This theory relies on representation structures called Discourse Representation Structures (DRSs), that are intended to capture the mental state of a reader as the discourse unfolds throughout the text. A DRS consists of a set of discourse referents representing entities which are in the discourse and a set of DRS conditions representing information about the referents. If we take the sentence "A man walked down the street" as an example, a DRS of this sentence would look like figure 2.3.


Figure 2.3: DRS of the sentence "A man walked down the street"

This representation can be seen as a variation of First-order predicate calculus. Nonetheless, one advantage of DRT is the treatment of anaphoric elements, as discussed in [25]. Anaphoric elements are distinct elements, usually pronouns, that map to the same semantic entity. For instance, if we expand the sentence of figure 2.3 to "A man walked down the street, and he saw a dog", the anaphora resolution task would be to state that "man" and "he" map to the same entity.

#### 2.1.2.3 Neo-Davidsonian representations

Neo-Davidsonian representations are representation that define verb predicates, as events and the thematic roles agent (or actor) and patient as the other terms modifying those events, as described in [42]. The colored terms in the previous sentence indicate that the next terms in this section with the same colors belong to the same-colored role.

This approach of considering that verb predicates stand for events and that the other predicates are modifying these events is very robust. It relieves the representation writer of creating verb arguments that encapsulate all the possible circumstances of the verb to just adding new arguments as modifiers of the verb. The prefix neo- indicates that all those modifiers are in the form of thematic roles.

For instance, the sentence (i) "Humans colonized the moon" would be parsed to (ii) "$\exists A, B, C \quad \text{moon}(A) \wedge \text{actor}(C, B) \wedge \text{human}(B) \wedge \text{theme}(C, A) \wedge \text{colonize}(C)$".

Consider the sentence (iii) "The moon was colonized by humans," this sentence is written in passive voice. Even though sentence (iii) is written in passive voice, it carries the same meaning as the sentence (i). This linguistic phenomenon is a common obstacle in language processing that is bypassed by neo-Davidsonian representations.

#### 2.1.2.4 Boxer pipeline

The Boxer pipeline consists of parsing the sentence with a CCG, then representing it in DRSs, and after that, the results are presented in a neo-Davidsonian framework [36].

Figure 2.4 shows boxer's output after processing it into a Logical Form. Predicates starting with 'pernam' (e.g., 'v1arrest') define the so-called thematic roles such as agent, theme, action etc., and other semantic roles such as person name (*pernam*) and even ad-hoc roles like *beach*. Every predicate (except for the person name one) is prefixed by its syntactic role as well.

Alice and Bob were arrested
yesterday in the beach.

pernambob(b),
pernamalice(a),
r1Time(v,y),
n1yesterday(y),
r1Theme(v,s),
v1arrest(v),
r1subset_of(b,s),
r1subset_of(a,s),
n1beach(p),
r1in(v,p).

Figure 2.4: Semantic Analysis by Boxer

In this dissertation, we apply those sentence parsers on the FrameNet annotated documents as part of our preprocessing step for the augmentation task. Each sentence is then converted to a set of annotated logical forms.

## 2.2 FrameNet

The definition of a frame, in Artificial Intelligence, is usually the one proposed by Minsky [33], a basic structure that underlies or supports a system, concept, or text. This definition of frame relates to the Frame problem, introduced by [32], in the concern about isolating domains of knowledge. The Frame problem emerges when we try to formalize representations of events that cause a change to a complex world, as described in [32] and [21].

This being said, we introduce a linguistic resource used throughout this dissertation, FrameNet. FrameNet consists of a network of structured frames, also known as FrameNet and a set of annotated documents; this resource is freely distributed by Berkley University.

### 2.2.1 FrameNet frames and frame elements

The FrameNet Project was started and primarily envisioned by Charles J. Fillmore in 1997 [16] with the purpose of providing a semantic representation resource that can be used by human linguists and by machines.

In the FrameNet lingo, frames stand for concepts like **Arrest**, **Coming to Believe** and **Event**. Those concepts can be seen as specific domains where entities take specific semantic roles, for instance, some of the roles an entity can take in the frame **Arrest** are Authority, Suspect and Place. Those semantic roles are called frame elements. Frames, among other things, hold a set of core frame elements and a set of peripheral frame elements. Each FrameNet frame has a description and a few examples. Those examples consist of annotated sentences that show the usage of the given frame and some of its frame elements in a real annotation; this directly maps to Minsky's definition of a frame.

### 2.2.2 Annotations

Besides the network of structured frames, the FrameNet Project also comprises a corpus of annotated documents, described as follow:

- The *network of frames* encompass a collection of frames where each frame element occurring in a frame has its own definition, written in human-friendly form. Those definitions usually carry an example sentence where the frame elements are annotated as well as the frame itself. So we have both frame annotations, also called

targets, and frame-element annotations. For simplicity, we are going to refer to frame-element annotations just as annotations for the rest of this dissertation.

- The *annotated documents* provided by the FrameNet project form a set of English written documents about a diverse range of subjects that were manually annotated by experts following the lexicon and structure of FrameNet. Thus, each sentence of each one of those documents was annotated in a three-step process; (i) if they had a word that linked the sentence to any frame this word is marked as a "target" word. (ii) if there are target words, those target words are related in the annotation to the frames that they betoken. After that, (iii) every sentence that has target words is then annotated with respect to the frame elements. Since every target word maps to a frame, the annotators then only annotate the sentence using frame elements from the frames pointed by a target in the sentence.

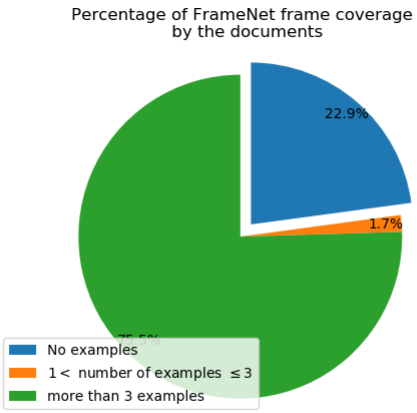



Figure 2.5: Lexical Augmentation

FrameNet is a widely used resource supporting a number of NLP tasks. However, as a manually-built resource, it is error-prone and incomplete. As shown in Fig. 2.5, we have observed that the frame coverage in FrameNet, that is, the number of frames that appear in at least one annotated sentence divided by the total number of frames, is only 70%. of the frames in the FrameNet do appear in the document annotations, as depicted in figure 2.5. More about FrameNet limitations and coverage is found in [35]. This paper describes five kinds of coverage gaps found in FrameNet. Some of those gaps are the absence of training data, i.e., labeled data similar to what was discussed above. The other kinds of deficiencies described in this paper, and stated to be more concerning by the author, are those of missing frames or target tokens (tokens from annotated sentences that relates those sentences to frames).

In this work, we intend to increase this coverage so that NLP tasks in general — and SRL in particular — are leveraged, by making more frame annotations available. If we can achieve some increase in frame annotations, even if it is not very large, it is bound to provide a relevant contribution to the Machine Reading community. That is because those annotated sentences feed in all Machine Reading pipelines that rely on FrameNet. For a rigorous and comprehensive description of the FrameNet project, we refer the reader to Fillmore et al. [16].

## 2.3 The Semantic Role Labeling (SRL) task

In this section, we revisit the Semantic Role Labeling (SRL) task, and how FrameNet supports it as a linguistic resource. We also discuss some of FrameNet's limitations from the SRL point of view. In doing so, we prepare the reader towards our specific research problem of augmenting FrameNet's semi-structured data, a subject that is more detailed in the next chapter, chapter 3.

The Semantic Role Labeling task in the NLP area consists of labeling the roles that entities took in a given sentence. In our previous example, "Humans colonized the moon", 'humans' would have the semantic role of 'colonizer', while 'moon' would be labeled as 'colonized place', depending on the underlying semantic role pool available. Other NLP tasks will have their own kind of annotations, in the case of tasks that produce labels, such as the part-of-speech tagging that classifies tokens into their part-of-speech class in the sentence.

This task motivates our semi-structured data augmentation problem. Since the state-of-the-art SRL parsers are statistical methods, and such methods rely on a good set of annotated sentences as examples, it is expected that an improvement on the annotation sets would lead improvement on the SRL parsers performance.

One possible source of SRL annotations is the FrameNet annotated document corpus. It consists of a corpus of textual documents whose sentences were annotated by humans. The general task of automatically generating frame-semantic annotations for an unseen sentence using this corpus as training data is called Frame-Semantic Parsing, FSP. This task has SRL as one of its three components, as follows.

Given a sentence, (i) *target identification* is the task of finding which token in the sentence should be matched to a frame; (ii) *frame identification* is to take this token and actually assign it to a specific frame; and (iii) *argument identification* is the semantic

role labeling task itself. It consists of matching frame elements that are members of the selected frames to the correct tokens in the sentence.

At the end of an FSP pipeline, the given sentence should have a set of target tokens from the tokens of the sentence, a set of frames, where each frame is associated with a target token, and finally a for each frame, it should have a set of frame elements associated to sequences of tokens in the sentence. For each of such steps, current state-of-the-art methods rely heavily on labeled data, i.e., manual annotations, both for training and for reporting their results.

FrameNet's set distribution of examples forms a long tail — a few frame elements have several examples over their related frames, while most of them have only one or none example at all —, making it difficult to tackle less popular frame elements. The importance of being able to parse the less popular frame rises when we tackle specific domains that make intense use of the concepts related to those frames and also to provide a fine-grained representation of the meaning of sentences. It means, in our case, more complete coverage of the semantic roles.

## 2.4 Open-Sesame: A semantic-role-labeling parser

Open-Sesame is a state-of-the-art method for frame-semantic parsing developed in [48]. This system is aimed at argument identification although it performs all the three Frame-Semantic Parsing tasks.

The Open-Sesame system is based on the segmental recurrent neural network, Seg-RNN, introduced in [30]. It was developed for handling the problem of segmenting sequential data — a generalization of the SRL problem, since the semantic roles are attributed to entities and those entities are represented in the sentences as spans of text. This model is useful in settings where the alignment between segments and labels is desired. To adjust the loss function to the argument identification task, [48] uses a softmax-margin cost function to favor recall.

Using a Seg-RNN, one is interested in learning the segments of a given sequence and also its labels, thus it leads to two possible learning tasks: (i) a fully supervised task where both the labels and the spans are known and (ii) a partially supervised task where only the labels are know

The architecture of the solution consists of a stack of bidirectional Long-Short Term

Memory neural networks, biLSTMs — biLSTMs are introduced in [43], they are a kind of Recurrent Neural Networks that can be understood as an attention mechanism for the model. The first layer is responsible for embedding a representation of the given sentence. At the end of the network, there is a final multilayer perceptron responsible for generating a segment factor that also takes as input an embedding representation of the frame related to the annotation.

The input of this network is a vector representing the sentence. Each token with position $q$ in the sentence will be represented by a vector $v_q = [d_q, e_q, o_q, \gamma_q]$. The vector $d_q$ is a learned embedding of the word type; $e_q$ is a pre-trained embedding of the word — the representation used was the GloVe embedding, introduced in [37]; $o_q$ is a learned embedding of the Part-of-Speech tag and $\gamma_q$ is the distance of the word to the beginning of the target. It does not rely on syntactic representations during the testing phase; this representation is used only during training. The training is done in a two-step approach, where the intermediary syntactic representation is used as a proxy for the first step. This way this system presents itself as a cheaper alternative — concerning computational resources and human effort — to SRL parsers, while stays a competitive approach to a more traditional pipeline.

Since there are much more spans of tokens that are not arguments (semantic roles) this work uses a cost function that favors precision over recall. The cost function used , defined in [18], is the softmax margin.

Our current augmentation method seeks to improve the overall performance of SRL parsers by providing to them more training data and, by doing so, mitigating this long tail effect. It is worth noting that if one intends to eliminate this effect entirely it would require a system that generates new sentences from scratch. This system should also be able to annotate the sentences, so at least it would solve the SRL task.

# Chapter 3

# Augmentation of FrameNet examples

The major contribution of this dissertation is our augmentation method that expands the argument labels (semantic role labels) available in the FrameNet project. We start the chapter by giving an overview of this method, and then we explore it in more details through a running example, then we present the method in more details.

## 3.1 The data augmentation problem

The FrameNet annotated documents set consists of annotated sentences, and those annotations comprise frame element annotations. On its turn, frame elements annotations can be seen as semantic role annotations, since they describe the semantic roles of the given entities in the sentence. This annotation structure was described in section 2.2.

The English FrameNet version 1.5 contains 1019 frames, and roughly 70% of them are depicted in the document annotations. These documents are used to train automatic Semantic Role Labeling (SRL) methods. There are also versions 1.6 and one 1.7 of the FrameNet, but they are still less commonly used in the literature. In particular, statistical methods thrive on a large pool of examples, and since each annotation is an example, generating new annotations of good quality is likely to improve the performance of such methods.

## 3.2 Our augmentation method

One way to add new annotations to less popular frames in the documents is to rely on the relationship among frames to migrate the annotations from one frame to the other.

The relationship among frames is explicitly stated in the FrameNet structure. Due to the similarity of frame element usages in those annotated sentences, we can create frame element annotations for a given frame by taking into consideration the annotations of related frames, called neighbor frames.

Our augmentation method consists of: given an annotation where some token spans are associated to frame elements of a specific frame, we replace the frame elements of the original with the frame elements of the neighbor frame, thus creating a new annotation taking advantage of the structural similarity of neighbor frames. This process is depicted in figure 3.1. The orange portion of the diagram shows the information used from FrameNet itself; we use it to point the neighbor frames and to extract the most common structure of its frame elements in example sentences. The green elements are the frame and frame elements extracted from the original sentence annotation. The blue part shows where the information of the original sentence is used. The yellow elements of the figure indicate what happens after we already have the intermediary representation of the sentence and the examples of the frame elements from both frames.

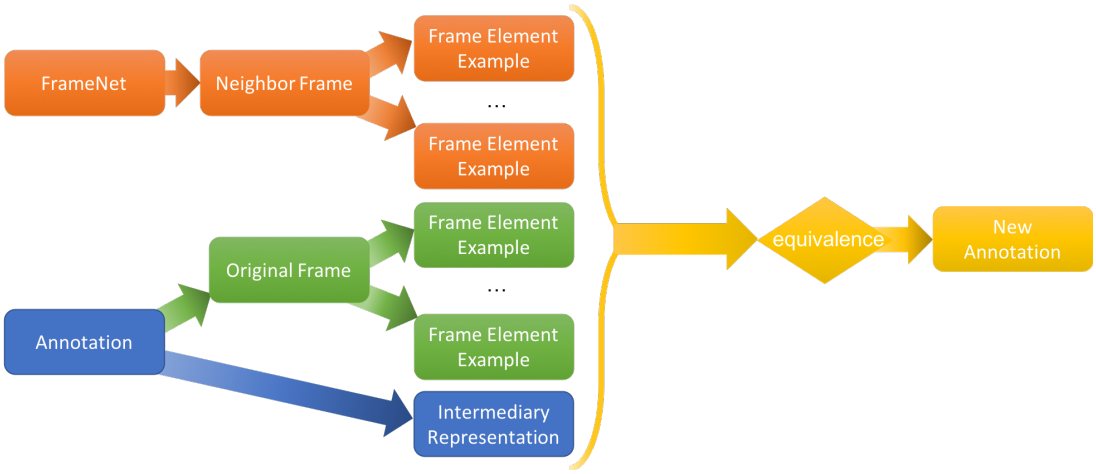


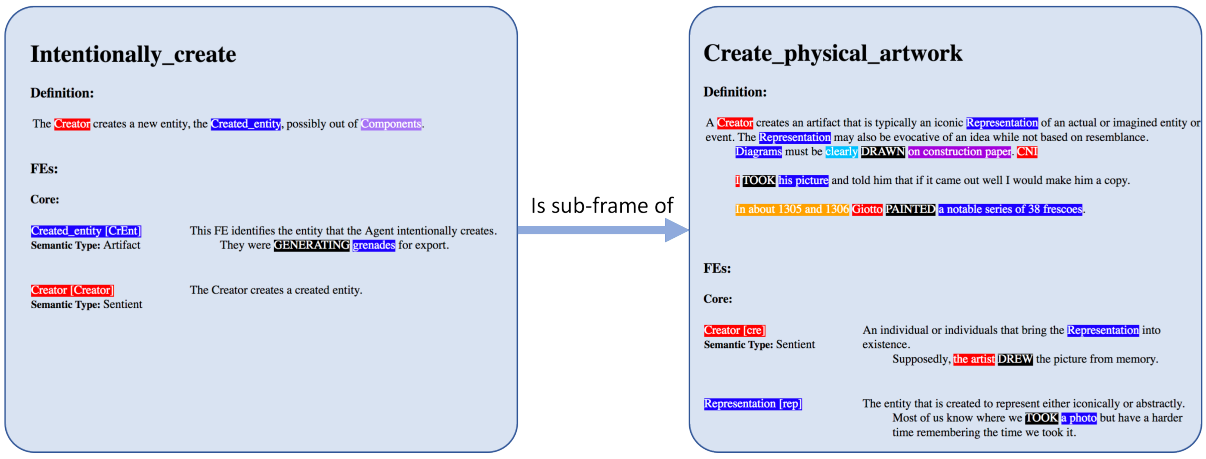

Figure 3.1: Augmentation method overview

Consider the sentence “We’ve found ways for people to enter the workforce”. This sentence is bound to the frame **Intentionally create**, i.e., this frame is the frame identified with this sentence. The annotation of this sentence concerning the structure of the frame **Intentionally create** is depicted in the figure 3.2a. There are two frame elements within this frame, namely, Creator and Created entity, which are mapped to the token spans highlighted with red and blue in the figure, respectively. The capitalized token span highlighted in black is the token span related to the frame.

From a general point of view, the data augmentation problem in this context is to ask how we could create a new annotation of this sentence — the present one is given

in terms of the frame **Intentionally create** — so that it could work as an annotated sentence example in terms of another frame as well.


(a) **Intentionally create** annotation with respect to the frame **Create physical artwork**




(b) Relationship among **Intentionally create** and **Create physical artwork** frames

Figure 3.2: Frames **Intentionally create** and **Create physical artwork**

Now consider **Create physical artwork**, another frame which is related to **Intentionally create** by the relation 'is sub-frame of', as shown in figure 3.2b. In this figure we see that **Create physical artwork** and **Intentionally create** have a definition section with a definition of their scopes in human-friendly terms. They also have a list of their frame elements, with examples of occurrences in a sentence. We exploit such inter-frame relations and then model the data augmentation problem accordingly. In our running example, the problem is reduced to whether or not we could build a new annotation of the sentence in terms of the structure of the frame **Create physical artwork** — that is, not only the frame itself, by means of the target token, but also its frame elements, namely, Creator and Representation. It is clear that "Ways for people with disability to enter the workforce" is an instance of **Intentionally create**, as this augmented annotation suggests.

### 3.2.1 Frame relations

The FrameNet inter-frame relations are the criteria we use to determine if two frames are "neighbors" in the augmentation method. We divide those relations into two sets:

(i) The set of *strong relations*, depicted in the table 3.1a, are the ones based in the *inheritance* and *part-of* concepts, and their reciprocal. (ii) The set of *weak relations* are the non-hierarchical relations. The set of strong relations seem to induce more reliable augmentations than the set of weak relations.

The relations that compose the strong and weak sets are listed in table 3.1, those relations are the ones that define a sense of hierarchy among frames, such as inheritance and *part-of* relations. This way, when we say that the frame **'Coming to believe'** inherits from **'Event'**, it means that **'Coming to Believe'** is an **'Event'** — **'Coming to Believe'** is the child frame and **'Event'** is the parent frame defined in this relation. So, when we say that a **'Halt'** is a subframe of *'Motion'* it means that the concept *'halt'* is part of the concept of *'motion'*.

| Relation | Description |
|---|---|
| Inherits from | is a frame of the same kind of the parent |
| Is Inherited by | the children frames have have the same kind |
| Subframe of | is a part of the parent frame |
| Has Subframe(s) | is composed by those frames |

(a) Strong relations

| Relation | Description |
|---|---|
| Perspective on | less strict inheritance |
| Is Perspectivized in | the children frames have a less strict inheritance |
| Uses | might be composed by those frames |
| Is Used by | might be part of the parent frame |
| Precedes | usually happens before |
| Is Preceded by | usually happens after |
| Is Inchoative of | the children are the cause of the root |
| Is Causative of | the root is the cause of the children |
| See also | Informational relation |

(b) Weak relations

Table 3.1: Inter-frame relationships

### 3.2.2 The notion of frame element equivalence

We model the frame elements equivalence concept regarding three different notions of equivalence: lexical, semantic and syntactic. We say that two frame elements from frames $X$ and $Y$ are lexically equivalent if they have the same name. The syntactic and semantic relations are based on the logical form representation of the frame element example annotations provided in the frame description. Those annotations are provided in the

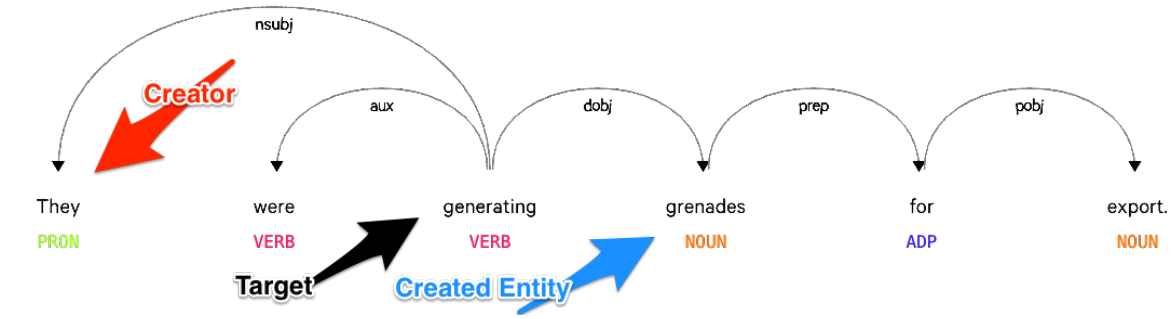
frame element descriptions from FrameNet and from the documents. Two frame elements are said syntactically equivalent if there is at least one pair of annotations from $X$ and $Y$ where these frame elements appear, and they have the same path of syntactic roles to the target in a syntactic representation. The semantic similarity follows the same concept of the syntactic equivalence, but, instead, we require a path of semantic roles in a semantic representation. The general equivalence procedure is illustrated in algorithm 1. Those three notions of equivalence further described in this section. The lexical notion is based in the comparison of frame element names and is already described in algorithm 1, the other two notions, syntactic, and semantic, illustrated in the algorithms 2, 3, and 3 respectively.

Let us recall the example depicted in figure 3.2b. In order to know if this annotation can be transferred to another frame **Create physical artwork**, we first have to check if all frame elements of **Intentionally create** in the annotated sentence are equivalent to some frame element in **Create physical artwork**. Using the notion of lexical equivalence, we consider Creator to be the same as Creator in **Create physical artwork** as they both have the same name. Using the syntactic equivalence we need to check if Created entity is equivalent to Representation. Each frame element is mapped to at most one frame of the other frame, thus Created entity could not be checked against Creator for equivalence.

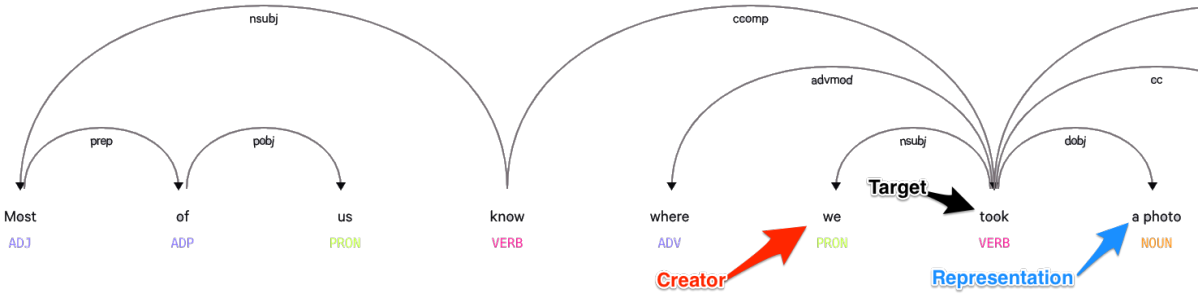
To that purpose, we take an example of Created entity from the **Intentionally create** frame and one example of Representation from the **Create physical artwork** frame and check if the syntactic path of the frame elements to the target of their frames is the same, as exhibited in figure 3.3. The syntactic representation of one example sentence of the frame element Created Entity from the frame **Intentionally create** is depicted in figure 3.3a. From that, it is easy to point a structure that relates the frame element to the target token, the relation 'dobj'. The same substructure is repeated in the syntactic representation of one example sentence of the frame element Representation, depicted in figure 3.3b. Thus, through the syntactic notion, they are equivalent.

Since each frame element in the annotation of **Intentionally create** is equivalent to some frame element in the annotation of **Create physical artwork**, we can copy this example to **Create physical artwork**. If there were any other frame elements left that have not an equivalent frame element in **Create physical artwork** using the lexical and syntactic equivalence notions, the next step would be to check their semantic equivalence the same way we did for the syntactic equivalence.

Those steps are detailed in algorithm 1. It is stated as a function eq_method that




(a) Syntactic representation of example in **Intentionally create**




(b) Syntactic representation of example in **Create physical artwork**

Figure 3.3: Syntactic representation of an example in the frame element descriptions

has as input two frames (the original frame of the annotation and the candidate frame), the list of frame elements from the original frame to be tested and the kind of equivalence notion to be used (lexical, syntactic or semantic).

**Algorithm 1:** Equivalence method

**Data:** Frames, frame1 and frame2;
List of frame elements in frame1 to be mapped to frame elements in frame2, from_fes;
List of kind of equivalence tests: lexical, syntactic or semantic

**Result:** Set of key-value pairs, where the keys are frame elements from the frame1 and values the frame elements of frame2

```
1  Function eq_method(frame1, frame2, from_fes, modes) is
2      fe_mapping ←∅;
3      not_mapped ← get_fes(frame1);
4      neighbor_not_mapped ← get_fes(frame2);
5      for fe in not_mapped do
6          for n_fe in neighbor_not_mapped do
7              n_graph = ∅;
8              if mode in "lexical" then
9                  if fe.name = n_fe.name then
10                     fe_mapping ← fe_mapping ∪ {(fe, n_fe.name)};
11                     not_mapped ← not_mapped \ {fe};
12                     neighbor_not_mapped ← neighbor_not_mapped \ {fe};
13             if mode in "syntactic" then
14                 n_graph ← get_dep_tree(neighbor_example) ;
15             if mode in "semantic" then
16                 n_graph ← get_boxer(neighbor_example) ;
17             if n_graph ≠ ∅ then
18                 n_paths ← paths_target_to_fe(n_graph, n_fe.name);
19                 if paths ∩ n_paths ≠ ∅ then
20                     fe_mapping ← fe_mapping ∪ {(fe, n_fe.name)};
21                     not_mapped ← not_mapped \ {fe};
22                     neighbor_not_mapped ← neighbor_not_mapped \ {fe};
23     return fe_mapping;
```

To test the equivalence of two sentence representation graphs, we compare all the non-cyclic paths between the target and the desired frame element on each graph, if there is at least one similar path in both graphs we say that those frame elements are equivalent, the

function for that is called paths_target_to_fe. The semantic representation is generated using the method get_boxer that is described in algorithm 3 in the next section.

The first step in our method after parsing the sentences using the syntactic or semantic parsers is to convert this representation into a logical form suitable for our subsequent processing and equivalence metrics. This logical form consists of first-order logic without any negation or quantifier, the quantifiers are removed through a grounding of the variables if needed. The variable grounding consists of replacing the free variables in the expression by constants.

From the syntactic parser we take the dependency tree and PoS tagging of the sentence. The output of the dependency tree is almost ready as our logical form. We use the top-level procedure described in algorithm 2 to generate the logical form. First, we rewrite every dependency relation $r$ of two tokens $v$ and $w$ as terms $r(v_i, w_i)$, where $v_i, w_i$ are unique constants respectively assigned to $v$ and $w$. We also rewrite every PoS tag $t$ of a given token $x$ as $t(x_i, x)$ terms, where $x_i$ is a unique constant assigned to $x$. Then we concatenate all those terms using 'and' operators. Finally, we add the predicate sentence_root($r_i$) for the unique constant assigned to the root token of the sentence.

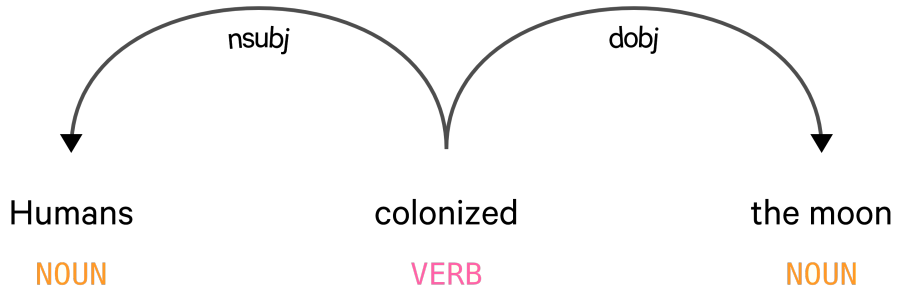
**Algorithm 2:** Syntactic representation of the sentence

**Data:** Function that generates a dependency tree of a given sentence, get_dep_tree;
Function that generates the PoS tags of the tokens of a given sentence, get_PoS;
Annotated sentence to be turned to a LF, sent
**Result:** Graph of the syntactic representation of sent

```
1  Function get_dep_tree(sent) is
2      sentence_dep_tree ← get_dep_tree(sent);
3      sentence_pos_tag ← get_PoS(sent);
4      lf ← list();
5      for r(v, w) in sentence_dep_tree do
6          v_i ← get_id(v);
7          w_i ← get_id(w);
8          lf.append( r(v_i, w_i) );
9      for t(x) in sentence_pos_tag do
10         lf.append( t(x_i, x) );
11     v ← sentence_dep_tree.root;
12     v_i ← get_id(v);
13     lf.append( sentence_root(v_i) );
14     sent_graph ← predicates_to_graph(lf);
15     return sent_graph;
```

If we take our example from the previous chapter, "Humans colonized the moon", its

dependency tree is depicted in the figure 3.4a and its logical form in the figure 3.4b.




(a) Dependency tree and PoS tagging of the sentence

$$\begin{aligned} \text{det}(\text{const2}, \textbf{`the'}) \ &\wedge \ \text{det}(\text{const2}, \text{const3}) \ \wedge \\ \text{noun}(\text{const0}, \textbf{`humans'}) \ &\wedge \ \text{nsubj}(\text{const0}, \text{const1}) \ \wedge \\ \text{noun}(\text{const3}, \textbf{`moon'}) \ &\wedge \ \text{dobj}(\text{const3}, \text{const1}) \ \wedge \\ \text{verb}(\text{const1}, \textbf{`colonized'}) \ &\wedge \ \text{sentence\_root}(\text{const1}) \end{aligned}$$

(b) Logical form obtained from the syntactic representation of the sentence

Figure 3.4: Logical form obtained and the syntactic representation of the sentence "Humans colonized the moon"

Although boxer's output is already provided in first-order logic [9], we still need to do variable grounding, followed by skolemization [2]. We also remove any negated terms and unbound variables left in order to have a simple graph structure. Every first-order formula may be converted into Skolem normal form while not changing its satisfiability through the skolemization process, this method consists of a successive application of equivalence rules to remove the existential quantifiers. Once we only have universal quantifiers, we are ready to ground the terms.

**Algorithm 3:** Semantic representation of the sentence

**Data:** Function that converts a sentence to FOL, sentence_to_fol;
Annotated sentence to be turned to a LF, sent
**Result:** Graph of the semantic representation of sent

```
Function get_boxer(sent) is
    sentence_fol ← sentence_to_fol(sent);
    skolem_fol ← skolemization(sentence_fol);
    grounded_fol ← variable_grounding(skolem_fol);
    lf ← negation_removal(grounded_fol);
    sent_graph ← predicates_to_graph(lf);
    return sent_graph;
```

### 3.2.3 Formalization

In order to make a general statement about our augmentation method, let us say **Create physical artwork** and **Intentionally create** are two frames $X$ and $Y$ linked by a given relation. As stated in section 2.2, there are two kinds of annotations in an annotated sentence, namely: target annotations and frame element annotations — the second kind we call annotation. Consider an annotated sentence $x$ with annotations of frame elements in $X$. Given that $X$ is related to $Y$ through one of the possible inter-frame relations (e.g., 'is sub-frame of'), we want to find what annotations we could extend to $Y$. That is, we want to know if there can be a new annotation of the sentence regarding the frame elements belonging to $Y$. So, we say that $x$ is transferable from $X$ to $Y$ if all the frame element annotations in $x$ are transferable to $Y$. We use this hard requirement that all frame elements are transferable assuming that using a soft constraint would result in new annotations that are less likely to be used interchangeably in the two frames.

An annotation is transferable from $X$ to $Y$ if each one of frame elements in $X$ is equivalent to one frame element in $Y$; we use three equivalence notions, lexical, syntactic, and semantic. This transferability assured, we can rewrite the sentence annotation using frame elements of $Y$, and we can add a new annotation to the sentence. Algorithm 4 brings the top-level procedure we follow to augment a given annotation.

**Algorithm 4:** The top-level augmentation algorithm

```
   Data: List of annotated sentences, anno_sentences;
   FrameNet instance, fnet;
   Function that realize the equivalence method used, eq_mode
   // eq_mode can be:  lexical, syntactic or semantic
   Result: List with the new annotations
 1 Function augment_annotation(anno_sentences, fnet, eq_mode) is
 2     new_annotations ← ∅;
 3     for anno_sent in anno_sentences do
 4         for frame in anno_sent do
 5             fes ← get_fes(anno_sent, frame);
 6             for neighbor in get_neighbors(frame, fnet) do
 7                 fe_mapping ← eq_method(frame, neighbor, fes, eq_mode);
 8                 if (fe_mapping ≠ ∅) ∧ (#fe_mapping = #f_es) then
 9                     new_annotation ← copy(anno_sent);
10                     change_frame(new_annotation, frame, neighbor);
11                     apply(fe_mapping, new_annotation, neighbor);
12                     new_annotations ← new_annotations ∪ {new_annotation} ;
13     return new_annotations;
```

Algorithm 4 depicts the function that takes a list of annotated sentences and augments them using the criteria described above. In this algorithm, an annotated sentence is an object that has the sentence and every annotation on this sentence indexed by the corresponding frame. In the algorithm, the operator "#" means the number of elements in the collection (e.g., number of items in a list, or the number of elements in a set). The function $get\_fes(annotated\_sentence,\ frame)$ returns the set of frame element annotations in the $annotated\_sentence$ that correspond to the given frame. The function $get\_neighbors(frame, framenet, relations)$ returns the set of the frames that are related to $frame$ through any relation listed in relations. The function $eq\_method(frame1, frame2, fes, eq\_mode)$ returns the mapping of the frame elements from $frame1$ listed in $fes$ to frame elements of $frame2$ using the $eq\_mode$ equivalence notion ($eq\_mode$ can be "lexical", syntactic, and "semantic"). This functions goes through every frame element in $fes$ comparing them to all the frame elements in $frame2$ that have not yet been matched as described in section 3.2.2. This method embodies the frame element comparisons that are used to decide if an annotation can be augmented or not.

In the next chapter, we describe the experiments that we carried out to evaluate the different strategies discussed above. We also show how the different relations and relation sets impacted the overall performance of the augmentation method; this performance is measured through the variation on the performance of the underlying SRL parser. In addition, we provide a description of the data used and their original coverage. These steps lead us to a discussion about the results observed, and they set the foundation for future work on chapter 6.

# Chapter 4

# Experiments

The purpose of the augmentation method we propose in this work is to increase the number of available examples and expand the coverage over less popular frames on annotated documents. This augmentation is particularly useful once we consider the difficulty in manually expanding the FrameNet example set and also the difficulty of, also manually, adding new documents. In the next sections we describe the methodology of the experiments performed in this dissertation, the data set we used, the results observed and some remarks about those results.

## 4.1 The data set used

Our dataset consists of annotated sentences from the collection of annotated documents made available in FrameNet release 1.5. This collection consists of 78 documents annotated by FrameNet's staff. Those documents hold together almost 6000 annotated sentences. In those annotated sentences is a total of almost 2400 frame annotations and more than 48000 frame element annotations related to those frame annotations. The prefix, that is, the part of the document name before '__' refers to the source of the document and the suffix is the document name. In total, there are more than 130000 sentences in the FrameNet project with some kind of annotation. More on the construction of this dataset and FrameNet, in general, is found in [17].

We divide those documents into three sets: train, validation, and test sets, as done with Open-Sesame in [48], and other work in the literature, such as [13]. We make it explicit in table 4.1, where the training set consists of all the documents that are not in the validation or test sets. That allows one to compare the results achieved here with related work relatively easily. We perform three strategies of augmentations namely, lexical,

syntactic and semantic. Section 3.2.2 holds a detailed description of those strategies. The gain on the overall number of annotations from each one of those strategies is depicted in figures 4.2b, 4.2c, and 4.2d respectively.

| | Document name |
|---|---|
| Test | ANC__110CYL067 |
| | ANC__110CYL069 |
| | ANC__112C-L013 |
| | ANC__IntroHongKong |
| | ANC__StephanopoulosCrimes |
| | ANC__WhereToHongKong |
| | KBEval__atm |
| | KBEval__Brandeis |
| | KBEval__cycorp |
| | KBEval__parc |
| | KBEval__Stanford |
| | KBEval__utd-icsi |
| | LUCorpus-v0.3__20000410_nyt-NEW |
| | LUCorpus-v0.3__AFGP-2002-602187-Trans |
| | LUCorpus-v0.3__enron-thread-159550 |
| | LUCorpus-v0.3__IZ-060316-01-Trans-1 |
| | LUCorpus-v0.3__SNO-525 |
| | LUCorpus-v0.3__sw2025-ms98-a-trans.ascii-1-NEW |
| | Miscellaneous__Hound-Ch14 |
| | Miscellaneous__SadatAssassination |
| | NTI__NorthKorea_Introduction |
| | NTI__Syria_NuclearOverview |
| | PropBank__AetnaLifeAndCasualty |
| Validation | ANC__110CYL072 |
| | KBEval__MIT |
| | LUCorpus-v0.3__20000415_apw_eng-NEW |
| | LUCorpus-v0.3__ENRON-pearson-email-25jul02 |
| | Miscellaneous__Hijack |
| | NTI__NorthKorea_NuclearOverview |
| | NTI__WMDNews_062606 |
| | PropBank__TicketSplitting |
| Train | *Remaining files* |

Table 4.1: Annotated documents split used in the experiments

One phenomenon perceived in the annotated documents that we aimed to mitigate in this work, is the long tail effect noticed when comparing the counting of annotations for each frame present in the documents, such effect is depicted in figure 4.1.

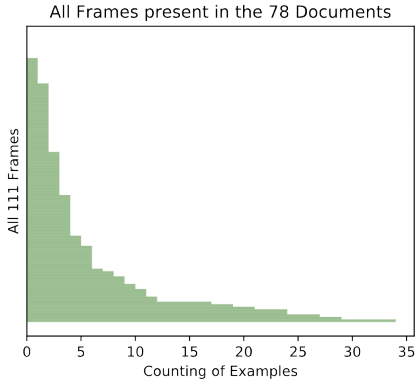



Figure 4.1: Examples per frames in the annotated documents, with no augmentation

## 4.2 Experimental methodology

In order to evaluate the augmentation strategies discussed in chapter 3 and to assess what kind of situations they work or not to improve an SRL parser, we run separated experiments for each one of the four different strategies of augmentations: lexical, syntactic, semantic and syntactic-semantic. The experiments consist of executing Open-Sesame [48], a state-of-the-art SRL parser, on each data set generated and then we compare the performance of it with the original data. We used each combination of augmentation strategy and relation set described earlier to generate different training and validation sets. This process encompasses the training of multiple Open-Sesame models that were tested and compared on the same test set.

The metrics of choice are precision, recall, and f1-score, for they are the most used in the literature for the SRL task and because they are well suited for a task where we want to focus on the positive cases. As done in previous works in the literature[3, 11, 12, 51], we take the micro-average of those sentences, i.e. we aggregate the metrics ignoring the division of sentences and documents found in the data. It means that, for every annotation, any time the parser predicts a frame element label and it is correct (i.e., the sentence token related to the annotation and the frame element are the same as the ones pointed in the gold standard[1]) we count it as a *true positive*; every time the parser predicts a wrong annotation; it is counted as *false positive*; and when the parser fails to predict an annotation that exists in the gold standard, it is counted as a *false negative*. Then we calculate the metrics over those total counts.

Each one of the multiple training instances is carried out until the same termination

[1]We call gold standard the set of manual annotations available in the dataset. They are used only for evaluating the method.

criterion is reached, for conformity and ease of comparison, the criterion is the same used in the Open-Sesame paper, we also used the default parameters reported in that paper [48]. This criterion is met when there where no updates in the best loss score reported after 28 validation epochs, although we noticed that convergence were reached much earlier. Convergence time was similar across each augmented data set and the original dataset, that dataset is further described in the next section, section 4.1.

We used the same GloVe embedding, [37] and optimized the model using ADAM, [27], with learning rate of 0.0005, the moving average parameter of 0.01, the moving average variance was set to 0.9999, and the $\epsilon$ parameter (to prevent numerical instability) was set to $10^{-8}$; no learning rate decay is used, as done in the original Open-Sesame paper.

## 4.3 Results

The impact of the augmentation method on the performance of the SRL parser is expressed in table 4.2. Values in bold are indicate improvement over the dataset without any augmentation, the values with an asterisk mark are the best values reported. We report precision, recall and f1-score metrics micro-averaged.This being said, our experimentation shows a moderate improvement on Open-Sesame's performance when trained on datasets that undertook the augmentation strategies developed here.

We see a growth of over roughly 13% of the original frame coverage using only the different kinds of augmentations. However, this augmentation method is not targeted to be used in the frame identification task since it introduces fake targets. Paramount any dataset augmentation process is the quality of the new data generated and how useful this data is to the methods that consume it. As discussed in the previous section, to evaluate the quality of the augmentation method, we investigate if there is an increase in performance of a model trained in augmented data over a model trained on the original data.

Besides testing different augmentation strategies, we group the relations used in this augmentations into Strong, Weak and All relations, as described in section 3.2.1. We also investigate the effect of each relation by itself. We see that figure 4.3 the syntactic augmentation using the strong relationships achieved best f1-score, but the same augmentation using only the precedence relation achieved the best precision.

We expected that the semantic would lead to better results over the lexical and syntactic strategies due to the robustness and refinement of the information provided by the

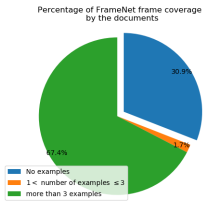
| | | Precision | Recall | $F_1 Score$ |
|---|---|---|---|---|
| | ALL | **0.5915** | **0.5724** | **0.5818** |
| | STRONG | **0.5839** | **0.5681** | 0.5759 |
| | inheritance | **0.5874** | **0.5591** | 0.5729 |
| | subframe | **0.5918** | 0.5486 | 0.5694 |
| Lexical | WEAK | **0.5884** | 0.5540 | 0.5707 |
| | precedence | 0.5704 | 0.5497 | 0.5599 |
| | causative | 0.5744 | **0.5756*** | 0.5750 |
| | use | 0.5683 | 0.5305 | 0.5487 |
| | perspective | **0.5923** | 0.5523 | 0.5716 |
| | ALL | 0.5787 | **0.5632** | 0.5709 |
| | STRONG | **0.6028** | **0.5631** | **0.5823*** |
| | subframe | **0.5907** | **0.5677** | **0.5789** |
| | inheritance | 0.5737 | 0.5335 | 0.5529 |
| Syntactic | WEAK | 0.5754 | 0.5488 | 0.5618 |
| | precedence | **0.6037*** | 0.5537 | **0.5776** |
| | causative | **0.5901** | 0.5465 | 0.5675 |
| | use | **0.5911** | **0.5611** | 0.5757 |
| | perspective | **0.5889** | **0.5672** | **0.5779** |
| | ALL | 0.5616 | 0.5207 | 0.5404 |
| | STRONG | 0.5729 | 0.5064 | 0.5376 |
| | subframe | 0.5703 | 0.5220 | 0.5451 |
| | inheritance | 0.5732 | 0.5334 | 0.5526 |
| Semantic | WEAK | 0.5665 | 0.5064 | 0.5347 |
| | precedence | 0.5430 | 0.4997 | 0.5205 |
| | causative | 0.5636 | 0.5295 | 0.5460 |
| | use | 0.5805 | 0.5062 | 0.5408 |
| | perspective | **0.5899** | 0.5296 | 0.5581 |
| No augmentation | | 0.5824 | 0.5567 | 0.5692 |

Table 4.2: Performance of Sesame with the different augmentations

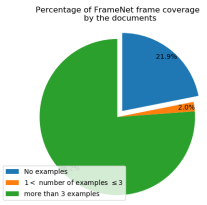
semantic parser. Nonetheless, we came to the conclusion that the semantic strategy was overcome by the lexical and syntactical ones.

Curiously enough, we perceived a non-additive effect of those relations, i.e., the combination of more than one relation can have worse results than the relations alone. It is clear when we compare the results of the Weak set of relations with the results obtained by each one of the relations in it, when considering the semantic and syntactic strategies. This is the case for the relation 'precedence'. That relation entails better results than the entire set of Weak relations, that contains this particular relation, when using the syntactic strategy.

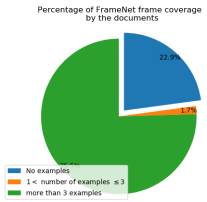
We theorize that this effect happens due to over-training caused by repeated or too similar annotations generated by the overlapping of the relations. The rationale behind



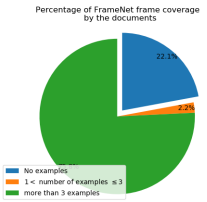

(a) No augmentation




(b) Semantic augmentation




(c) Lexical Augmentation




(d) Syntactic augmentation

Figure 4.2: Augmentation frame coverage

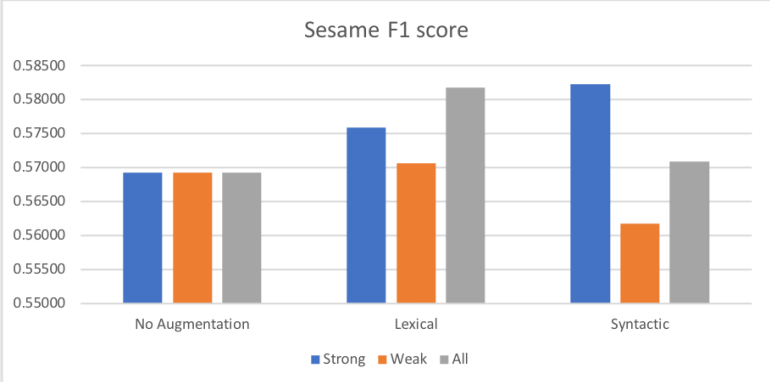



Figure 4.3: Comparison of Sesame $F_1$ Score

that phenomena is that some relationships will culminate in similar new annotations, thus leading the model to over-train. For instance, many annotations have only an Agent in it, and usually the target token is a verb with the tokens marked as Agent its subject. And some frame transferences by using the frame relations are not very meaningful. For instance, take the example used in chapter 3, "We've found ways for people to enter the workforce". It is an example of a good and meaningful transference, but the same syntactic method we used to generate this example also transfers wrongly this sentence annotation to **Create Physical Artwork**.

Further analysis is needed for a better understanding of why the semantic strategies did not perform well on this evaluation, but manual inspection suggests that it is due to the inaccuracy of the semantic parser tested. The semantic parser failed to parse many complex sentences and gave many imprecise parsing results for the sentences it was able to parse. Sentences that are more than 110 characters long constitutes 43.6% of all the


(a) **Intentionally create** annotation with respect to the frame **Create physical artwork**


(b) Example of a new annotation transfered to **Achieved First** frame elements

Figure 4.4: Annotation transference example between **Intentionally Create** and **Achieved First**

sentences in the document corpus, and for the majority of those sentences, the semantic parser failed to give a reasonable parsing or did not return a parsing at all.

The sentence "Far more progressive than the Archaics, the Anasazi utilized such formal agricultural techniques as irrigation to assist their harvest.", figure 4.5a, extracted from the document 'ANC__HistoryOfLasVegas' is an example improper semantic parsing.


(a) Sentence from 'ANC__HistoryOfLasVegas', that is not correctly parsed by the semantic parser, with it's original annotation

noun(c0,'harvest') ∧ relation(c0,c1,'of') ∧ noun(c1,'thing') ∧
**place(c2)** ∧ **noun(c2,'anasazi')** ∧ **place(c3)** ∧
**noun(c3,'anasazi')** ∧ place(c4) ∧ noun(c4,'archaics') ∧
relation(c6,c12,'while') ∧ topic(c6,c5) ∧ theme(c14,c0) ∧
actor(c14,c3) ∧ verb(c14,'assist') ∧ theme(c6,c10) ∧
actor(c6,c2) ∧ verb(c6,'utilize') ∧ relation(c10,'such') ∧
relation(c10,c7,'as') ∧ noun(c7,'irrigation') ∧ noun(c10,'technique') ∧
adjective(c8,'agricultural') ∧ theme(c8,c10) ∧ adjective(c9,'formal') ∧
theme(c9,c10) ∧ relation(c12,'far') ∧ relation(c12,c4,'than') ∧
adjective(c11,'more') ∧ relation(c12,c11,'manner') ∧ theme(c12,c13) ∧
adjective(c12,'progressive') ∧ noun(c13,'thing').

(b) Boxer parsing after turning this sentence into logical form, with one parsing mistake highlighted using bold type

Figure 4.5: Example of semantic parsing mistake that induces an incorrect annotation

Two major problems with this parsing involve the entity 'Anasazy': (i) it is duplicated because a failure to handle the anaphora present in the text, and (ii) this entity is labeled as an Agent, but it receives a 'place' predicate which when considering the frame elements

of frames related to ‘Using’, might be used to mistakenly classify it as the frame element Place, as depicted in figure 4.5b. None of those problems were caused by the logical form conversion, but were observed in Boxer’s raw output as well.

Besides that, one can also argue that the logical form method construction used incurs in information loss that might further undermine the semantic parsing results, see section 3.2.2.

In conclusion, we achieve a moderate improvement of the performance on the SRL task by employing this augmentation, that is we not only increased the number of annotations to the most popular frames but were able to add annotations to the less popular ones. The long tail problem, i.e., the fact that many frames in the annotated documents are present in fewer annotations — in opposition to a minority of frames that occur in many annotations —, still is a challenge to be overcome. This problem of augmenting FrameNet without any other linguistic resource, however, does not seem to be suited to be solved by an automatic augmentation strategy, at least for now. But yet, it seems to need a careful selection of more documents to be annotated. Those new documents should be selected in such a way that they are more likely to cover the less popular frames. Nonetheless, our augmentation is able to add meaningful sentence annotations to the overall set of frames improving this coverage.

# Chapter 5

# Literature Review

In this chapter, we discuss the existing literature more related to our work. We considered the three main areas that we have built our contribution upon on, namely: Sentence Representation, Semantic Role Labeling, and Linguistic resources augmentation.

## 5.1 Logical form and sentence representation

Textual data is found in unstructured ways, as mentioned throughout this dissertation, and we want to make it as structured as possible, so it is machine processable. By machine processable, we mean that automatic methods would be able to query for the information contained in the text, combine it with information from other sources and query this knowledge. Logical forms can be used to express both the syntactic and semantic aspects of the sentences of a textual document, and much work have been done on building such logical forms. It is a usual step to parse a sentence into a syntactic representation and use this intermediary representation to generate a semantic representation of the meaning covered in the sentence.

In particular, [38] devises a system based on the lambda calculus for deriving neo-Davidsonian, see section 2.1.2 logical forms from dependency trees (a kind of syntactic representation, as explained in section 2.1.1). They evaluate the quality of such logical forms derived from the dependency trees of the sentences by feeding those logical forms to a semantic parser. This semantic parser consists of a graph matching algorithm that matches the structure of the logical form to Freebase, a collaboratively created tuple-based knowledge base that later on was used to power the Google's Knowledge Graph initiative, [45]. It generates a robust representation of the sentences and can be compared with our current approach in future work. Using this approach as our semantic parser would be a

promising comparison since one of their claims is that this representation outperforms a CCG-based representation which composes the Boxer method, used in our work.

Similarly to our work on generating a logical form is the work found in [38]. They create a new neo-Davidsonian representation of sentences that might improve our current method. There is also the work done in [6] combines logical and distributional representations. It uses similarity metrics to create weighted rules using Markov Logic Networks [39]; they show that besides estimating similarity between sentences, this method can also recognize textual entailment. Such textual entailment could be used as another feature for our augmentation purposes.

In the same way, we rely on Boxer to obtain a logic-based parsed output. Previous work has already started from this tool to extract and represent meaning in a structured, machine-processable format from text documents. In particular, [6, 7] combined the parsed logical representation with distributional semantics and Markov Logic Networks. The distributional semantics is used to construct a unified knowledge base from different sources, while MLN is used to perform inference. The neo-Davidsonian representation and MLN are also employed to solve the Science and Math challenge, an NLP competition that aims to produce systems that can answer fifth-grade science exams, as done in [26].

One key component of our augmentation method is the comparison of similar substructures of sentences. We compare sub-sentences described in logical forms to find if two frame elements from two different frames have a similar semantic role in at least one of the example sentences available in the FrameNet's graph of Frames. There is a rich literature in sentence similarity, most of it focuses on structural similarity and also entailment similarity, i.e., the similarity between the underlying meaning of the sentences.

The difficulties on directly applying those methods without any tinkering to our problem are that we calculate if substructures in the sentence are similar focusing on specific terms. It is not clear how to apply this concept to most of those methods since those methods are not concerned with specific terms of the sentence, but the sentence as a whole. An interesting survey of different similarity methods is found in [20]. This survey segments them into three approaches: string-based, corpus-based and knowledge-based similarity methods.

## 5.2 Semantic Role Labeling

The Semantic Role Labeling is the problem of finding semantic roles to entities located in textual documents, and we refer the reader to section 2.3. SRL is a rich area of research containing work that takes advantage of multiple linguistics resources including FrameNet. The most recent and state-of-the-art approaches are mostly based on statistical methods, in particular, machine learning methods.

The model presented in [12] uses latent variables and semi-supervised learning to improve frame disambiguation for targets unseen at training time. On the other hand, the work shown in [22] consists of a frame identification that is coupled into an argument parsing method to perform FSP. Sling, [40], is a framework for frame-semantic parsing that performs neural-network parsing with bidirectional LSTM input encoding and a transition based recurrent unit. It takes as input only the tokens of the sentence, skipping any previous syntactic or semantic parser. Both methods are machine-learning based.

The semantic parser developed in [19] connects VerbNet and FrameNet by mapping the FrameNet frames to the VerbNet Intersective Levin classes. To further increase the verb coverage they use the lexicon contained in PropBank; they also use the semantic annotations in the PropBank dataset for evaluating their system.

## 5.3 Linguistic resources augmentation

To the best of our knowledge, this is the first work that builds a data augmentation strategy relying on only the data provided by FrameNet. Other venues of work combine other linguistic resources with FrameNet. To produce SRL parsers, [44] [19] are examples of work that combine different linguistic resources, PropBank and VerbNet, with FrameNet.

The model proposed in [34]is based on word embedding to identify a mapping between Wikidata relations — Wikidata is a free knowledge base aimed to be read by humans and machines and further discussed in [49] —, and FrameNet frames and to annotate the arguments of each relationship with the semantic roles from the second resource. It is a case where FrameNet is used to enrich other resources and is a clear contrast with our work that aims to enhance FrameNet without the use of external corpora, but only on parsing methods. This choice makes this approach flexible and agnostic of external data sources used to train those parsers.

# Chapter 6

# Final Remarks

Natural Language Processing (NLP), is a wide research area that encompasses many subareas concerned not only with the syntax of the written language but also with its semantics pragmatics and discourse motives. The importance if this area ranges from better human-machine interfaces to automatically constructing knowledge bases from the vast amount of textual data available nowadays. Such diverse area defines a number of tasks that are supposed to constitute a pipeline of language processing.

On the semantics branch of NLP, we have the task of attributing semantic roles to entities with respect to the sentence or small portion of text at a time. This task is called Semantic Role Labeling (SRL), and it is an essential task towards creating a representation of textual information that is easily processed and consumed by machines. Among the available resources that can be used for assisting or evaluating SRL methods, FrameNet is one of the most well-known and widely used. However, as a manually-built resource, it is error-prone and incomplete. A large group of frames lacks useful annotations, both in the annotated documents made available by the FrameNet project and also in annotated examples in the Frames and Frame Elements descriptions. In this work, we presented and evaluated a data augmentation method for FrameNet documents that increases by over 13% the total number of annotations.

For the best of our knowledge, most work on the literature on augmenting FrameNet is concerned with combining it with other linguistic resources such as PropBank and VerbNet. Our work is intended to take information from the FrameNet and use it to expand the annotations found in the corpora of annotated documents that FrameNet provides.

## 6.1 Our method

Our augmentation method consists of a preliminary parsing of the annotated sentence; this parsing might be just lexical, syntactic or semantic; then we convert this representation in a logical form that is suitable for our equivalence check. We can transfer an annotated sentence with respect to a frame $f$ if there is any frame $f'$ related to $f$ that has frame elements equivalent to each frame element in the original annotation; this way we create a new annotation concerning $f'$. This equivalence is verified against the parsing of example annotated sentences found in the description of the frames and frame elements themselves. Our augmentation is particularly suited to create more examples for the SRL task.

After applying this augmentation the total of unique frames covered by the document annotations increased by over 41%, which translates in a frame coverage of over 13%. The meaningfulness of the augmentation for the proposed task was evaluated in chapter 4.3; we say that a set of annotations is meaningful if it helps to improve the results of an automatic parser.

Our method induced an improvement on the results of a state-of-the-art SRL parser, which indicates that data augmentation for semantic annotations can be further explored to improve the performance of current parsers. The increase in performance coupled with the knowledge that specific relations seem to lead to better results point to a promising venue of work from where any statistical method for SRL can benefit from, as discussed in section 4.

As a result of our work, a new dataset is now available for SRL and frame-semantic parsing in general. The code for the generation of the augmented documents is open (under MIT license) and can be found at `http://github.com/lorel-uff/srl-nlp`.

The results obtained with the augmented datasets indicate that one might want to experiment different ways to harvest the information from the syntactic sentence representations, or couple the linguistic resource with other available resources, such as VerbNet and PropBank. One could also try to infer new links among frames to boost the augmentation algorithm. Not forgetting to mention that our augmentation method is focused on one specific task, and two more tasks are usually associated with FrameNet annotated document set. Those possible venues of work are further explained in the next section.

However, as discussed in chapter 4, our current augmentation method is not targeted to be used in the frame identification task since it introduces fake targets. At the moment that we create a new annotation based on related frames, we have an equivalence of the

frame elements. However, we do not have an equivalence of the targets. Because of that, we replicate the targets. This replication inserts fake targets, and it might harm the performance of automatic frame identification methods.

## 6.2 Future work

The research done in this project showed that it is possible to automatically expand some of the content in the FrameNet documents to improve the performance of state-of-the-art SRL parsers. This very finding leads to other questions like how to improve such augmentation, and what other kinds of augmentation can we try.

One immediate future work is to analyze the behavior of other SRL parsers when trained using data from different augmentations strategies. Other statistical methods might have different behavior from what we observed in this work. With this consideration in mind, we already started to explore symbolic methods for the SRL task. Such models have the advantage of being explainable, and their internal representation can be used to explicit patterns and characteristics found in the annotations. Besides that, we expect to be able to better understand the effects of our augmentations strategies by comparing different models.

Many other intermediary sentence representations can be explored within our augmentation methods, like the representation developed in [38], which is a good example of a possible new representation to be tried in our augmentation pipeline. The only thing to keep in mind, considering our current augmentation pipeline, is whether this sentence representation can be searched to provide a meaning for a Frame Element associated to this sentence. That representation can be used then to compare the Frame Elements to others.

Another aspect of the linguistic resource that can be studied in the future is the connection among frames. One possible unfolding of such study would be the use of link prediction algorithms, as some of the ones explored in [23], to expand the inter-frame relationships available in FrameNet.

We also intend to test the method on other electronic (linguistic) resources. For example, WordNet seems a relatively close opportunity for short-to-mid-term research. A less immediate opportunity is to work with semi-structured data from other electronic resources. This way one could exploit the structure of Wikipedia infoboxes as a sort of annotated content towards extracting semantics from their associated articles and combine

it with Fillmore's Frame Theory.

Looking further into the literature, one can propose to use PropBank and VerbNet together to help in the augmentation process. PropBank and VerbNet are linguistic resources that were already used in conjunction with FrameNet for the SRL task, we refer the reader to chapter 5 for more details about this literature. The main contribution of those other resources to the SRL task would be: a new set of examples from PropBank and more information to be derived from the sentence. This information can be used to compare and match two different Frame Elements and improve our augmentation pipeline.

We also consider how to augment the annotation sets targeting the other two tasks, target identification and frame identification, mentioned in section 2.3. As quickly described earlier, those two tasks are steps that precede argument identification, i.e., SRL. In this scenario, a robust SRL parser, when introduced to a new sentence with no prior annotations, would have to rely on a previous system that already identifies the target (the word in the sentence that is associated with some frame) and also identifies the correct frame. Armed with this information the SRL parser would be able to find the entity roles — the frame elements — in the sentence.

At last, all this work in SRL is also intended to be used in further work on Open Domain Question Answering. Semantic representations of pieces of text and reasoning on those representations seem to be essential in this setting. It is due to the complex reasoning that can be involved in fully representing the intent of a query. A robust and yet sound semantic representation of the underlying semantics of a sentence, coupled with extensive common sense knowledge, can be used to feed a reasoning system for processing such queries.

Our work in this dissertation shows that it is possible to transfer annotations across frames in order to augment a linguistic resource to improve the performance of state-of-the-art SRL parsers and it indicates a wide range of possible research venues on this topic.

# References


[1] Abend, O., Rappoport, A. The State of the Art in Semantic Representations. *Acl 35*, 3 (2017), 23–24.

[2] Baaz, M., Leitsch, A. On skolemization and proof complexity. *Fundamenta Informaticae 20*, 4 (1994), 353–379.

[3] Baker, C., Ellsworth, M. SemEval ' 07 Task 19 : Frame Semantic Structure Extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations* (2007), no. June, p. 99–104.

[4] Baxter, R., Hastings, N., Law, A., Glass, E. J. Combinatory Categorial Grammars: Generative Power and Relationship to Linear Context-Free Rewriting Systems. In *Proceedings of the 26th Annual Meeting on Association for Computational Linguistics* (1988), p. 278–285.

[5] Belew, R. K., Belew, R. K. *Finding out about: a cognitive perspective on search engine technology and the WWW*, vol. 1. Cambridge University Press, 2000.

[6] Beltagy, I., Chau, C., Boleda, G., Garrette, D., Erk, K. Montague Meets Markov: Deep Semantics with Probabilistic Logical Form. *Joint Conference on Lexical and Computational Semantics (*SEM) 1* (2013), 11–21.

[7] Beltagy, I., Roller, S., Cheng, P., Erk, K., Mooney, R. J. Representing meaning with a combination of logical and distributional models. *Computational Linguistics 42*, 4 (2016), 763–808.

[8] Bird, S., Loper, E. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* (2004), p. 31.

[9] Bos, J. Wide-coverage semantic analysis with Boxer. *2008 Conference on Semantics in Text Processing*, c (2008), 277–286.

[10] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research 12* (2011), 2493–2537.

[11] Das, D. Statistical Models for Frame-Semantic Parsing. *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, 2007 (2014), 26–29.

[12] Das, D., Chen, D., F. T. Martins, A., Schneider, N., Noah A. Smith, N. Frame-Semantic Parsing. *Computational linguistics 40*, 1 (2014), 9 –56.

[13] Das, D., Schneider, N., Chen, D., Smith, N. A. Probabilistic Frame-Semantic Parsing. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL) 3*, June (2010), 948–956.

[14] Eijck, J. V. Discourse Representation Theory The Problem of Unbound Anaphora. *Encyclopedia of language and linguistics 3* (2005), 660–669.

[15] Etzioni, O., Banko, M., Cafarella, M. J. Machine reading. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference* (2006), AAAI Press, p. 1517–1519.

[16] F. Baker, C., J. Fillmore, C., B. Lowe, J. The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1* (1998), p. 86 – 90.

[17] F. Baker, C., J. Fillmore, C., Cronin, B. The Structure of the FrameNet Database. *International Journal of Lexicography 16*, 3 (2003), 281—296.

[18] Gimpel, K., Smith, N. A. Softmax-margin CRFs: Training log-linear models with cost functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2010), p. 733–736.

[19] Giuglea, A.-M., Moschitti, A. Semantic Role Labeling via FrameNet, VerbNet and PropBank. *Proceedings of the 21st International Conference on Computational Linguistics*, July (2006), 929–936.

[20] Gomaa, W. H., Fahmy, A. A. A Survey of Text Similarity Approaches. *International Journal of Computer Applications 68* (2013), 13–18.

[21] Hayes, P. J. The frame problem and related problems in Artificial Intelligence. *Readings in AI* (1981), 45–59.

[22] Hermann, K. M., Das, D., Weston, J., Ganchev, K. Semantic Frame Identification with Distributed Word Representations. *Proceedings of ACL* (2014), 1448–1458.

[23] Huang, D., Trotman, A., Geva, S. Experiments and evaluation of link discovery in the wikipedia. In *Proceedings of the SIGIR 2008 Workshop on Focused Retrieval* (2008), Citeseer, p. 22–29.

[24] Kamp, H. A theory of truth and semantic representation. *Formal semantics-the essential readings* (1981), 189–222.

[25] Kamp, H., Reyle, U. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory.* Springer Science \& Business Media, 2013.

[26] Khot, T., Balasubramanian, N., Gribkoff, E., Sabharwal, A., Clark, P., Etzioni, O. Exploring markov logic networks for question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015* (2015), ACL, p. 685–694.

[27] Kingma, D. P., Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[28] Kingsbury, P., Palmer, M. From Treebank to PropBank. *LREC* (2002), 1989–1993.

[29] Kipper, K., Korhonen, A., Ryant, N., Palmer, M. A large-scale classification of English verbs. *Language Resources and Evaluation 42*, 1 (2008), 21–40.

[30] Kong, L., Dyer, C., Smith, N. A. Segmental Recurrent Neural Networks. *arXiv preprint arXiv:1511.06018* (2015), 1–10.

[31] Màrquez, L., Carreras, X., Litkowski, K. C., Stevenson, S. Semantic role labeling: an introduction to the special issue, 2008.

[32] McCarthy, J., J. Hayes, P. Some Philosophical Problems from the Standpoint of Atificial Intelligence. *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: demo papers* (1969).

[33] Minsky, M. A framework for representing knowledge, 1974.

[34] Mousselly Sergieh, H., Gurevych, I. Enriching Wikidata with Frame Semantics. *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, 3 (2016), 29–34.

[35] Palmer, A., Sporleder, C. Evaluating FrameNet-style semantic parsing: the role of coverage gaps in FrameNet. *Computational Linguistics*, August (2010), 928–936.

[36] Parsons, T. *Events in the Semantics of English*. Cambridge, Ma: MIT Press, 1990.

[37] Pennington, J., Socher, R., Manning, C. Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), 1532–1543.

[38] Reddy, S., Täckström, O., Collins, M., Kwiatkowski, T., Das, D., Steedman, M., Lapata, M. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the ACL 4* (2016), 127–140.

[39] Richardson, M., Domingos, P. Markov logic networks. *Machine learning 62* (2006), 107–136.

[40] Ringgaard, M., Gupta, R., Pereira, F. C. Sling: A framework for frame semantic parsing. *arXiv preprint arXiv:1710.07032* (2017).

[41] Roth, M., Lapata, M. Neural Semantic Role Labeling with Dependency Path Embeddings. *arXiv preprint arXiv:1605.07515* (2016).

[42] Schubert, L. Computational Linguistics. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., spring 201 ed. Metaphysics Research Lab, Stanford University, 2015.

[43] Schuster, M., Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing 45*, 11 (1997), 2673–2681.

[44] Shi, L., Mihalcea, R. Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing. *Computational Linguistics and Intelligent Text Processing* (2005), 100–111.

[45] Singhal, A. Introducing the Knowledge Graph : things, not strings, 2012.

[46] spaCy Team. spaCy: Industrial-strength NLP.

[47] Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics* (2012), Association for Computational Linguistics, p. 102–107.

[48] Swayamdipta, S., Thomson, S., Dyer, C., Smith, N. A. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *arXiv preprint arXiv:1706.09528* (2017).

[49] Vrandecic, D., Krotzsch, M. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM 57* (2014), 78–85.

[50] Wikipedia contributors. Wikipedia: Size of wikipedia, 2018. [Accessed: October 15, 2018].

[51] Yang, B., Mitchell, T. A Joint Sequential and Relational Model for Frame-Semantic Parsing. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017), 1258–1267.