Natural Language Generation in the Logos Model

Sara Amato
Independent Scholar, Italy
saraamato11@gmail.com

Kutz Arrieta
Independent Scholar, USA
kutzaki@gmail.com

Abstract

In this paper, we focus on the generation module in the Logos Model and, more generally, target modules via generation-specific linguistic challenges, illustrating them with examples taken from Italian and Spanish as target languages. We briefly explore the different models and applications in existence for Natural Language Generation as context for the description of the Logos Machine Translation Model.

1 Introduction

Natural Language Generation has a long tradition in the field of Computational Linguistics. It can be defined as the means and methods to produce human language, be it from another language, from coded instructions, from graphical representations or from datasets. The modules to be included in a generation component will vary greatly depending on the methods used to produce natural language.

Much has changed in the field since the Logos Model was active as a commercial system. It is beyond the scope of this paper to provide a review of those changes. The release of Large Language Models (LLMs) to the public in the last few months is shifting the research and development paradigm for Natural Language Processing and Generation. There is much to say about LLMs. Here we just want to bring the reader's attention to the term "Generative AI". In its most basic sense, Generative AI (Artificial Intelligence) is a type of artificial intelligence technology that can produce various types of content including text, imagery and audio. It produces synthetic data (computer-generated content). This is exactly what Natural Language Generation does. What differs is the methodology. In this paper we focus on a methodology that has nothing to do with Generative AI in its current meaning.

There are products, applications and research prototypes that deal with the task of generating human language from data. Some of these have as their final product biographies or résumés; others produce reports of different types. One aspect that is shared by these different Natural Language Generation (NLG) applications is that some type of parsing (natural language processing) is involved. One doesn't go from data "straight" to generation. Instead, data needs to be analyzed for relevance and classified, and then facts and factoids (in Natural Language Processing (NLP) factoids are small information units about the world) need to be extracted. In this view, the data needs to be understood before any kind of language generation can occur.

Other initiatives have tried to convert schemata or different types of graphic representations into human language. In this case also, the schema needs to be understood first, to be "parsed," before generation occurs. Between the steps of parsing and generation, several other modules might be present, such as sentence planners or tag classifiers. These modules are usually preceded by information planning modules.

Here, we are focusing on language generation in the context of Machine Translation, more specifically, in the Logos Model. The Logos Model is mostly based on linguistic knowledge, both syntax and semantics, supported by semantic and world knowledge encoded in a knowledge graph and a relational database.

2 Types of Generation

As mentioned in Section 1, generation is realized in different ways depending on the restrictions or potential of the system at hand, and on the application for which it is intended. In the remainder of Section 2 we mention some of these applications.

Sections 3 and 4 describe generation in the Logos Model through examples. We conclude in section 5.

2.1 Applications

Generation is used to build well-formed sentences from basic meaning components. From something like *someone has children*, the user chooses from options such as sentence focus, gender of the speaker, etc. This is the case of Phrasomatic, for example.

Generation may be used to create language models that are easily understood by humans not requiring them to have specific knowledge of a certain domain. Håkan Burden and Rogardt Heldal, in the context of Model-Driven Engineering, have experimented with the use of Natural Language Generation to go from a Platform-Independent Model (PIM) to a Computational Independent Model (CIM) using Grammatical Framework. The result is a description of the original software model as well as the underlying motivations for design decisions, in the form of natural language texts.

Generation may be used to generate reports from data sets, such as in the proposal from Arria. The idea is to save time for users who need to analyze large amounts of data, such as finance portfolios.

Generation may be used to produce both questions and explanations from Natural Language understanding and reasoning systems. This is the case of products such as KnowMatters or IBM Watson.

For all the cases aforementioned, different approaches to natural language are used. Some are unification-based solutions, such as Tree-Adjoining Grammars (TAGs), etc., but we will not go into detailed descriptions of these systems, since our goal here is to describe the Logos approach to generation.

2.2 Models

In Machine Translation (MT) systems based solely on statistical or neural models, there is no, or very limited, semantic generation. Statistical MT systems match patterns in aligned bilingual texts to build a statistical model of translation. This has nothing to do with the tasks of parsing and generation in systems based on linguistic knowledge.

Dependency Grammars have been and are being used in several models. In these models, sentence generation is viewed as a sequence of transductions (surface representations), produced by different grammars.

The Universal Networking Language (UNL) has also been used as a tool in generation systems. Logos uses its own semantico-syntactic abstraction language (SAL) throughout its modules.

In some systems, generation starts with some type of logical representation by projecting a "general" syntactic structure. After this, generation rules apply and produce the desired output in the target language.

We should mention here some methods and projects which are, akin to the Logos Model, knowledge- and rule- based such as the Wikimedia Abstract Language Project. In this project LLMs are not being adopted because the main goal is to make it possible for less-resourced languages to generate content and the assumption is that those languages do not have enough digital content for the models to be trained on. Worth mentioning here also is the work from Maria Keet and her team in University of Cape Town on isiZulu languages.

3 Natural Language Generation in Logos

The Logos model is described in detail elsewhere (Scott 2003 et al.). We will not repeat such detailed descriptions here, but we include just enough of how the Logos model works to better understand where and when generation happens.

We should mention, though, that most of what has been written about the Logos model deals with source analysis. Very little has been written about its Generation module, often referred to as Target Generation.

There seems to be some kind of "exhaustion effect" when it comes to generation: tokenization, resolution, lexical matches, relation to source syntactic and semantic parsing, etc. Several extremely important things need to happen before going into Generation. But the application has to decide clause boundaries and dependencies, resolve ambiguities, group phrases, understand phrase dependencies, etc., in order to provide the Generation module with the most precise information possible.

Everyone in MT is aware of the importance of generation. After all, it is what the user first sees: how "good" the system is at producing a language that mimics native speaker abilities. But the next and more important factor for the user is how close the target is to the source, how faithfully it reflects the information provided in the source language. In addition, there is a good variety of editing tools, and thus, it makes sense to put most of your energy in source analysis and provide the Generation module with just what it needs to produce an acceptable output that can easily be edited. So, Logos adopted the correct strategy in regards to the distribution of resources in the model.

Nevertheless, greater independence between source and target modules was being planned to make it perfect in every possible aspect and to increase its modularity.

3.1 Description

The Generation or target module is described in Scott 2003, but let's list here its main features.

The TRAN label refers to phases in the syntactic and semantic parsing and generation pipeline. Generation does not start in TRAN4, even though it is considered the most "generation-like" TRAN. TRAN4 is the final stage in the generation pipeline. In TRAN1 we already have rewrite rules (rules that transform source language structures into target language structures). Some of the behaviors of generation rules are directly controlled by source analysis rules. Source analysis triggers generation rules. While rewrite rules occur early in the translation process, they are not considered "pure" generation. "Authentic" generation rules occur in TRAN4. These rules are quite complex (often more so than source rules). TRAN4 builds well-formed sentences in the target language. TRAN4 and, therefore, the generation module, is supposed to be multi-source, and should not depend on the particular source analysis of one specific language or another. It is based on an abstract representation or interlingua.

The semantico-syntactic (SAL) representation that Logos uses to encode languages and rules is an important asset for Generation. In the end, the Logos model has proved very successful in understanding that a higher level of abstraction is required when coding and classifying parts of speech, which goes beyond the usual part of speech classifications (nouns, adjectives, etc.). The Logos classification, based on this higher level of abstraction, reflects something that we could call the deep semantic functionality of each part of speech, whereby different members of a word class belong to a similar semantic category provided that they trigger similar syntactic behavior: *send* and *give* have identical chains of semantico-syntactic codes (manually assigned in the knowledge graph) because a) their deep semantics calls for a second indirect object and b) the indirect object can be introduced by the preposition *to* when following the direct object (*he gave a camera to his wife*) or by no preposition at all by inverting the order of the two objects (*he gave his wife a camera*). The verb *communicate*, instead, shares only part of its semantico-syntactic chain with *send* and *give*, because its syntactic environment only shares with them a) but not b). When looking at Logos SAL coding we see a representation that mimics what happens in our brains when processing natural languages.

Target rules are part of the generation module. Target is produced incrementally. Morphological and semantic information in the lexicon often encodes features needed in target generation. We would like to highlight here that the morphological modules in the Logos model, even though seldom described, are a great feature of the model. In some sense, the morphological modules in Logos are also "mixed" in the sense of parsing and generation. These modules need to encode all the information necessary to function for both a source and a target language. In parsing systems the morphological modules only need to take into account analysis cases; they don't need to restrict "overgeneration," as it is assumed that spurious tokens should not occur in the input. When a morphology module is to be used for both parsing and generation, the rules need to be much more precise to avoid spurious tokens in both directions. When building a morphology module for a parsing-only system, rules can generalize surface token to lemma rules, assuming the spurious surface form will not appear in text, and if it did, it would not morph to any valid lemmas.

The challenges when building morphological components for parsing or for generation are quite different. The fact that the design of the model allows for the morphological components to be used in both directions leaves very little room for "free rides", i.e. situations where possible counter-cases do not arise, such as morphology parsing, where spurious forms would not be part of the data to be parsed.

For example, in a morphology for analysis one could write a rule for any clitic and any number of clitics to be attached to a Spanish verb in the gerund or infinitive form, assuming a text written in Spanish will not have combinations not allowed. This is to say that in the process of parsing one can assume that no spurious combinations will be present. While, if the morphology is to be capable of generating forms in Spanish, more complex rules must be written to allow only grammatical combinations of clitics and verbs and prevent over-generation.

In the transfer phase, parse and generation, the source tree is built and rebuilt through its source analysis while accommodating the needs of the target language. The Logos Model uses TRAN rules. These are syntactic rules rooted in the semantics of the components or entities. TRAN rules are target or group-specific, and they call target-specific tables

(30-tables, 40-tables and 50-tables). These tables accomplish different tasks, getting more and more specific to a given target as the pipeline progresses. After this, the generation phase takes place, where constituent movement, lexical selection and final formatting take place.

3.2 Evolution of the Model

The Logos model has been evolving since its conception. As mentioned before, the last phase of this evolution was aiming at a greater separation of source and target modules.

It has not been easy to classify Logos among the MT systems because of the partial separation of parse and generation, the transfer modules, the shared semantic rules and the use of an abstract representation (SAL). For its design to move toward a full interlingua model, source and target language need to be more independent of each other while maintaining the complex lexicon structure and SAL language, which allows for a semantico-syntactic representation of knowledge through Natural Language representation. This change in design is motivated by the need for modularity in order to improve results and to accelerate the addition of new language pairs. This change was started but has not been completed. In this new design all source operations are completed independently of the target language, and target languages need only to concentrate on generation from a SAL parse tree, without any concern of impact on the source language parse or on other target languages.

4 Linguistic Challenges in Generation

In this section we discuss some of the challenges Generation modules face. These are challenges that any system needs to address and solve in order to produce the correct results.

4.1 Verb Phrases and Verb Compounds

Generation of verb compounds and phrases is addressed in SemTab rules that are specific to a language pair. SemTab is explained elsewhere (e.g. Scott 2018); here we see some examples where SemTab rules handle “verb + particle” structures.

1. LOOK (VI) OUT (PART) = TENER CUIDADO
2. LET (VT) OFF (PART) = DESPEDIR

After the RES (resolution) module has resolved that an element is a particle and not a preposition or an adverb, for example, the combination of the "verb + particle" strings in a rule represents a different verb, with a different semantico-syntactic code from the one assigned to the original main verb, and a different transfer in the target language.

4.2 Semantic Context

In the case of the “verb + noun rule” exemplified here, we are taking a set of nouns that belong to a certain semantic category and handling the combination of the copulative verb and any of these nouns, under any form or any modification context, as an idiom. Therefore, the translation should be tailored to the target language.

3. BE (VI) (UNITS OF LINEAR MSR-PREC BY ARITHMATE) = MEDIR N

In German source, separable verb prefixes and particles must be reassigned to the verb so that they can be handled as a single string. In a sequence like: *Wir drehen weiter* each word enters the translation module separately. Therefore *drehen* and *weiter* would, by default, be handled separately. Once RES confirms that *weiter* should be treated as a separable particle there will be a match on rules like:

4. DREHEN WEITER = CONTINUARE A GIRARE

This rule re-codes the verb *drehen* as the verb *weiterdrehen* to allow a match on another very generic SemTab rule coded for *weiterdrehen*, which will generate the appropriate translation in Italian.

5. WEITERDREHEN N = CONTINUARE A GIRARE N

This module, even though not a part of Generation per se, is a very elegant way to handle these types of transformations.

4.3 Adverb Generation: Form and Position

Adverbs play an important role and are often difficult to generate correctly. They have syntactic scope, therefore, their position in the target sentence is syntactically relevant and they take different shapes.

6. EN - errantly

   ES - de manera errante

   IT - a casaccio

In the case of the adverb in example 6, we do not want to generate the default *errantemente* through the lexicon and/or the morphology in every case. *-ly* adverbs in English cannot be treated equally, depending on their semantics and their position in the sentence, the Generation module needs to treat them differently. Adverbs such as *roughly, generally* and *slowly* do not belong to the same semantic category. *Slowly* is the default case as *-ly* (or *-mente*) derivationally creates adverbs of manner from adjectives, while *roughly* is more a modifier than an adverb of manner.

4.4 Quantifiers

Parsing quantifiers presents serious challenges. Generating the appropriate quantifiers in the target language is not trivial. Quantifiers are another example of several semantic and syntactic complex issues in which the design of the parse has to either be "complete" or take into account the needs of the target languages.

7. EN - *any two books*

   ES - *dos libros cualesquiera*

The default transfer for this phrase would have been **cualquier dos libros*, but a TRAN rule, dealing with the source noun phrase analysis and sending a signal to the transfer module causes the Generation module to effect the correct output. Therefore, in these cases, as in many, source analysis and target generation are intertwined.

4.5 Clitics

Pronominal clitics in Romance languages are extremely difficult to handle in an NLP application. By comparison with other systems, Logos performs very well, as all the information needed to choose between **le** and **con é**l, etc. in different contexts is provided in the source analysis.

8. EN - *You may contact him*

   ES - *puede contactarle*

The Logos Model produces: ES**:** *se puede poner en comunicación con él*

As we see in the example both outputs are correct, as *puede contactar con él* would have been, but it is a challenge to decide which should be the default strategy: attached clitic or preposition + pronoun?

In this specific example, a SemTab rule is making the decision:

9. CONTACT (VT-ACTIVE) N (NOM-HUMAN) N = N PONERSE(REFL) EN COMUNICACIÓN CON N

We should note here that Logos in its design allows for very creative and productive strategies. The "black hole" strategy, initially conceived for dealing with clitics in Spanish, is a good example of this. For example, a verb in English might be translated by a verb phrase in Spanish. For example, *to stock → almacenar en el sótano*. If you decide in Generation to attach your clitic at the end of the verb phrase, you would get ungrammatical outputs such as **quiero almacenar en el sótanolo* because the system sees the string *almacenar en el sótano* as the verb transfer in Spanish and attaches the clitic at the end.

There were several ways this could be handled in-house, but, since Logos allowed its users to have proprietary dictionaries, the question of how to solve this in a systematic and predictable way arose. Every verb phrase of more than one word in Italian or Spanish may have a black hole, and the Generation rules ask the verb: "Do you have a black hole?" If true, the clitic goes into the black hole (located just after the head of the verb phrase). If false, it attaches at the end of the verb. This results in huge improvements for Generation. These black holes can also be used in noun phrases, adjectival phrases, etc.

Let us consider the English verb *ask,* which is translated in Italian by the verb *chiedere.* You may decide to attach the clitic at the end of the verb like in *ask him → chiedetegli* or to place the pronoun before the verb at the beginning of the clause like in *you may ask him → gli potete chiedere.* When the clitic is loaded at the end of the verb phrase, and the verb phrase is complex, the exact same behavior described in Spanish occurs in Italian: *you can always give it to your teacher → lo potete sempre dare al vostro insegnante.*

4.6 –ed in English

Another big group of Generation challenges are the *-ed* verb forms in English and their translation in Spanish and Italian.

10. EN - *The file is displayed by John*

    ES - *John visualiza el fichero*

    IT - *John visualizza il file*

11. EN - *The file is displayed by clicking the mouse*

    ES - *Se visualiza el fichero chasqueando el ratón*

    IT - *Si visualizza il file cliccando sul mouse*

12. EN - *English is spoken here*

    ES - *Aquí se habla inglés*

    IT - *Qui si parla inglese*

English makes a very different usage of resultatives and passives as compared with Romance languages. The Generation module has to decide if the appropriate outcome is to transform the sentence into its active counterpart, maintain a passive or generate an impersonal sentence, among others. If the source parsing doesn't carry enough information (information that may not be needed for parsing per se), the Generation module cannot make the correct decision. The Logos Model handled these challenges well.

TRAN4 through its rules and tables determines if the noun phrase that follows *by* is really an agent and sends a signal. If it is an agent, the outcome in Spanish will be an active sentence (*John visualiza el fichero*). If it is not, it will be rendered as an instrumental in the target (**Se visualiza el fichero chasqueando al ratón*). Note the incorrect *al* in this sentence, probably due to a rule too powerful dealing with accusative animate complements in Spanish.

In other cases, and again through signals, in this case adapted to the needs of the target, the system will output an impersonal sentence (*Aquí se habla inglés*).

Italian exhibits similar behavior:

13. EN - *The file is displayed by John*

    IT - *John visualizza il file*

14. EN - *The file is displayed by clicking the mouse*

    IT - *Si visualizza il file cliccando sul mouse*

15. EN - *English is spoken here*

    IT - *Qui si parla inglese*

4.7 ser and estar / essere and stare

Spanish and Italian have two verbs to be, *ser* and *estar*, *essere* and *stare*. Deciding which one to use presents a great challenge for human learners of the language. To encode this distinction in a Generation system is as much of a challenge. For this, Logos implements a strategy that makes use of almost every module in the system. This is another one of those cases where the distinction between source analysis and target generation is really blurry. The Generation module needs great amounts of information from the source to make the decision. This information is not actually needed for the parse and it might not be needed for other target languages. Therefore, this need is encumbering the source analysis modules with a considerable amount of additional work.

We are ignoring here the idiomatic cases where the English verb is to be is translated by a completely different verb in Spanish (*be five meters long → medir cinco metros*).

16. EN - *I am dead*

    ES - *estoy muerto*

Choosing the verb *estar* in Spanish occurs in SemTab, before TRAN4. *ser* and *estar* rules in TRAN4 will check if there has been a match in SemTab and the issue has been solved. In that case, TRAN4 will not do anything.

17. BE (VI) ADJ (DEAD) = ESTAR ADJ (MUERTO)

    EN - *I am a dead horse*

    ES - *soy un caballo inactivo*

Source analysis knows that *dead* is modifying *horse* and not referring to the subject and, therefore the SemTab rule won't apply. TRAN4 runs all the necessary checks to make sure *ser* is the correct choice.

18. EN - *I am yellow*

    ES - *soy amarillo*

In TRAN4 the conditions for *estar* are not met, it is a basic predicative adjective, therefore, we chose the default case: *ser*. But, as we know, both are possible, but have different meanings (*soy amarillo* and *estoy amarillo*), but without any further modification in the sentence (*estoy amarillo de rabia*) or contextual information, the correct call is to use *ser*.

19. EN - *I am tired*

    ES - **se me cansa*

The correct output would be *estoy cansado*. Note that *tired* is a verbal adjective. Yet another example of the dependency between source and target. This small sentence is not analyzed correctly in the source and it is nearly impossible for the Generation module to recuperate from this. It should be noted that most of these nearly idiomatic cases are easily handled nowadays in other models such as statistical machine translation.

4.8 Existentials

Existentials, such as *there is* or *there are* in English are well known MT challenges.

20. EN - *There are toys here*

    ES - *Hay juguetes aquí*

21. EN - *There are broken toys here*

    ES - **está roto los juguetes aquí allí*

    ES - *Hay juguetes rotos aquí*

The system tries to match in SemTab rules such as:

22. BE ADV (HERE) = ESTAR ADV

But TRAN4 signals the system that we are dealing with an existential and therefore, *hay* must be produced. In the second case, this interaction fails and the system already in TRAN3 has misidentified the *–ed* (*broken*) as a resultative and *there* as a spatial adverb. Even though the model exhibits a great deal of flexibility by which one can recuperate from incorrect parses, it is not always done.

Not all shortcomings in the Logos Model should be understood as limitations of the design or the technology. If it had been an academic system maybe we would expect it to accommodate academic quality measure requirements, but it was a commercial system and, therefore, the measures of goodness are different. A commercial system is concerned with efficiency, cost, time to market, etc., while an academic system is not.

4.9 Ellipsis and other Special Cases

Sometimes the source languages allow certain ellipses that the target might not. The missing components have to be retrieved. These issues can be easily fixed in Logos.

23. EN - *If necessary*

    ES - *si fuera necesario*

Apparently harmless lexical entries such as *just* can stir a great amount of trouble. In some cases it is just an adverb, in others it is part of a verbal structure that needs to be rendered as such in the target. In this case the distinction is probably necessary for both source parse and target generation. It is certainly indispensable for the correct target generation.

24. EN - *I just arrived*

    ES - *acabo de llegar*

    IT - *sono appena arrivato*

25. EN - *It's just late*

    ES - *es simplemente tarde*

As can be observed, SAL comes in handy, as the distinctions have to do with the different types of adjectives and adverbs. This semantic typology is captured in the SAL language. Therefore, making use of the power of the SAL code, these issues can be resolved.

4.10 Adjective Ordering

When a noun in the source language is modified by more than one adjective, one needs to make decisions on the order these adjectives should follow in the target language. Via TRAN4 rules, the Logos Model encodes ordering restrictions for adjectives. This is not a major issue, but possibly one that creates editing work for translators and is easily solved in target tables.

4.11 Elision in Italian

In Italian, the final vowel of a determiner must be elided in certain contexts. It is handled by the so-called Finish Rules. It is an orthographic pattern which applies to Italian articles and demonstrative adjectives (*uno/una*, *il/lo/la/i/gli/le*, *quello/quella*) when the following word begins with a vowel (e.g., *uno albero* → *un albero*). In certain cases an apostrophe is added (*lo albero* → *l'albero*; *quello albero* → *quell'albero*; *una opera* → *un'opera*; *la opera* → *l'opera*; *quella opera*→ *quell'opera*).

Once the whole translation module has assigned the appropriate transfers and gender settings, Finish Rules will provide the correct spelling adjustments.

4.12 Determiners

A known nightmare in Spanish and Italian generation is the presence or absence of determiners. It seems like an impossible issue to solve at a reasonable cost. Logos does not do well with determiners, but then again, no one does. This is a difficult generation issue to solve and often the approach is to post-edit the incorrect translation rather than generating it.

Logos is a commercial system and, when a development team is deciding what issues to tackle, several factors come into play. Two very important factors for any commercial Generation system are comprehensibility and ease of edition. Generation needs to produce an output that is easily understood (and, of course, faithful to the source), and, if it needs to be edited (often the case with Machine Translation), how easily is the output edited? How many strokes? How many words?, etc.

The case of the determiners in Spanish and Italian is representative of these concerns. For a native speaker it is extremely easy to fix the presence or absence of determiners, and determiners are small words. This explains why a strategy for determiners in Spanish and Italian Generation in the Logos MT system has not been a priority.

5 Conclusions and Future Work

In this paper we have briefly presented Natural Language Generation in its broad sense and the main models and applications that utilize Generation. We have described Generation in the context of the Logos Model. We also provided some examples and raised some relevant questions in the field of Natural Language Generation.

The logical next step in the Logos Model is Target Independent Analysis (TIA). As mentioned earlier, this will allow for modularity and independent linguistic work. But TIA will have to offer an intermediate system where additional source analysis operations might be performed for the sake of the Generation module. Generation needs information to make decisions, and that information must come from somewhere, ideally, from an Interlingua that faithfully and abstractly represents the input. As an example of the consequences of this separation, target SemTab and target verb valence information could be encoded, providing the Generation module with very powerful tools.

From a broad point of view, the Logos Model should probably find a way to integrate statistical and neural models into its rule-based system. Combining the power of these strategies could make the Logos Model the best performing system in the market. Designing and implementing such integration is no easy task, and is beyond the scope of this paper.

References

Anand, Tej andKahn, Gary. 1992. *Making Sense of Gigabytes: A System for Knowledge-Based Market Analysis*. A. C. Nielsen Company. IAAI-92 Proceedings, San Jose, CA, USA.

Arria - http://www.arria.com/science.php

Bach, Nguyen. 2012. *Dependency Structures for Statistical Machine Translation*. Ph.D. Dissertation. Carnegie Mellon University. Pittsburgh, PA, USA

Barreiro, Anabela, Scott, Bernard, Kasper, Walter and Keller, Bernd. 2011. *Open Logos Rule-Based Machine Translation: Philosophy, Model Resources and Customization*. Machine Translation 25, pp 107-126.

Håkan Burden and Rogardt Heldal. 2011. *Natural language generation from class diagrams*. Proceedings of MoDeVVa Proceedings of the 8th International Workshop on Model-Driven Engineering, Verification and Validation. New York, NY, USA

Gdaniec, Claudia. 2002. *Lexical Choice and Syntactic Generation in a Transfer System Transformations in the New LMT English-German System*. In Machine Translation and the Information Soup. Volume 1529 of the series Lecture Notes in Computer Science pp. 408-420.

Lareau, François and Vanner, Leo. 2007. *Towards a Generic Multilingual Dependency Grammar for Text Generation*. Proceedings of the GEAF 2007 Workshop.

Orliac, Brigitte and Dillinger, Mike. 2003. *Collocation extraction for machine translation*. Proceedings of Machine Translation Summit IX: Papers. New Orleans, USA

Phrasomatic - http://www.phrasomatic.net/

Scott, Bernard E. 2018.Translation, Brains and the Computer: A Neurolinguistic Solution to Ambiguity and Complexity in Machine Translation. Machine Translation: Technologies and Applications, vol 2. Springer, Cham.

Scott, Bernard and Barreiro, Anabela. 2009. OpenLogos MT and the SAL representation language. In Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, pages 19–26, Alacant, Spain.

Scott, Bernad E. 2003. *The Logos Model: An Historical Perspective*. Machine Translation 18: 1–72. Hingham, MA, USA

Scott, Bernad E. 1992. *Competence, Performance and the Paradigm Shift: a Connectionist Perspective*. Logos Corporation Technology Center, Mount Arlington, NJ.

Scott, Bernard E. 1990. *Biological Neural Net for Parsing Long, Complex Sentences, Logos Corporation Technical Report*. Mount Arlington, NJ, US