Multilingual Online Translation

Non multa, sed multum

Contract No.:	FP7-ICT-247914
Project full title:	MOLTO - Multilingual Online Translation
Deliverable:	**D5.3 WP5 final report: statistical and robust MT**
Security (distribution level):	Public
Contractual date of delivery:	M38
Actual date of delivery:	2nd April 2013
Type:	Regular Publication
Status & version:	Final
Authors:	Cristina España-Bonet, Ramona Enache, Krasimir Angelov, Shafqat Virk, Erzsébet Galgóczy Meritxell Gonzàlez, Aarne Ranta, Lluís Màrquez
Task responsible:	UPC
Other contributors:	UGOT

ABSTRACT

This is the final report of *Workpackage 5: Statistical and Robust MT*. The document includes the work done during the period lasting since M24, when D5.2 was submitted. Contributions correspond on the one hand to the improvement of the previous hybrid MT systems, its portage to German, and the development of new hybrid systems. On the other hand, the generation of general lexical resources and robust parsing are introduced for the first time. Previous work reported in *D5.1 Description of the final collection of corpora*[a] and *D5.2 Description and evaluation of the combination prototypes*[b] has been summarised here in order to make the document self-contained.

[a] http://www.molto-project.eu/biblio/description-final-collection-corpora

[b] http://www.molto-project.eu/biblio/deliverable/description-and-evaluation-combination-prototypes

Contents

1 Introduction

This is the final report of *Workpackage 5: Statistical and Robust MT*. The goal of the workpackage has been to develop translation methods that complete the grammar-based ones to extend their coverage and quality in unconstrained text translation. This does not only concern the translation engines but also individual components and resources that are relevant in a translation engine.

From the work done within this framework, we highlight the generation of lexical resources from WordNet [13], Apertium dictionaries [14], and SMT translation tables; the development of a statistical robust parser; and hybrid systems for patents that combine an in-domain GF grammar with an SMT translator. The first two points allow to extend the coverage of GF and are useful for a general translation or a translation in any domain. The third one, on the contrary, starts from the translation on a concrete domain and tries to extend the coverage outside the coverage of the grammar.

The work introduced here corresponds to the development of the components. It is not a stand-alone work but it needs of other workpackages in order to be integrated within MOLTO. As input for the patents related work, we use data coming from the use case, WP7, where also the engines are finally used. As outcome of the workpackage, we also provide WP3 with new translators to integrate and the work done is being disseminated within WP10. The relation with WP9 is bidirectional during the whole project since a continuous evaluation is necessary to develop the engines.

The deliverable is organised as follows. After this introduction, Section 2 is devoted to show the distinctive features of patents as one of the main domains of application of the technology. The data and the necessary pre-process are also reported. Section 3 describes the statistical translation and evaluates automatically its performance. The same is done in Section 4 for the GF translation system for patents. The general architecture is described as well as the particularities of the French and German grammars. Next, in Section 5, a general methodology for extracting lexicons from existing resources is explained. The robust statistical parser is reported in Section 6 and the construction of the hybrid engines for patents in Section 7. Finally, Section 8 relates this work to others inside and outside MOLTO and Section 9 summarises the work. We also include two appendixes. Appendix A includes the automatic evaluation of the hybrid systems of Section 7. The number of variants of the systems is too high and the complete evaluation within the main text would damage the readability of the document. Appendix B is a short manual for using the hybrid system developed for patent translation.

2 Patents Translation

Patents have been chosen for the opening of the system to non-restricted language. This election has two main reasons. First, the language of patents, although having a larger amount of vocabulary and richness of grammatical structure than the mathematical exercises and the description of museum objects, still uses a formal style that can be interpreted by a grammar. And second, there is nowadays a growing interest for patents translation. The high and increasing number of registered patents has created a huge multilingual database of patents distributed all over the world. So, there is an actual need for building systems able to access, search and translate patents, in order to make these data available to a large community. Hence, the translation of patents text seems a natural scenario to test the techniques developed in this workpackage.

2.1 Patent documents

A patent is an official document granting a right. The file or files associated to every patent contain not only the terms of the patent itself, but also bibliographic data (i.e. publication, authorship and classification). Being an official document, the structure giving the terms of the patent is quite fixed. The documents are normalised to an XML format, in which the standardised fields include dates, countries, languages, references, person names, and companies as well as rich subject classifications. Every patent has a title, a description, an abstract with a short and general summary and a series of claims.

The *technical data* is a relevant section of the document. It contains the list of IPC[1] codes assigned to the patent, i.e. the classification of the patent according to the different areas of technology to which the patent pertain. The IPC is arranged in a hierarchical structure of 8 sections divided into 120 classes, 600 subclasses and 70,000 groups. Here we show the first level of the classification:

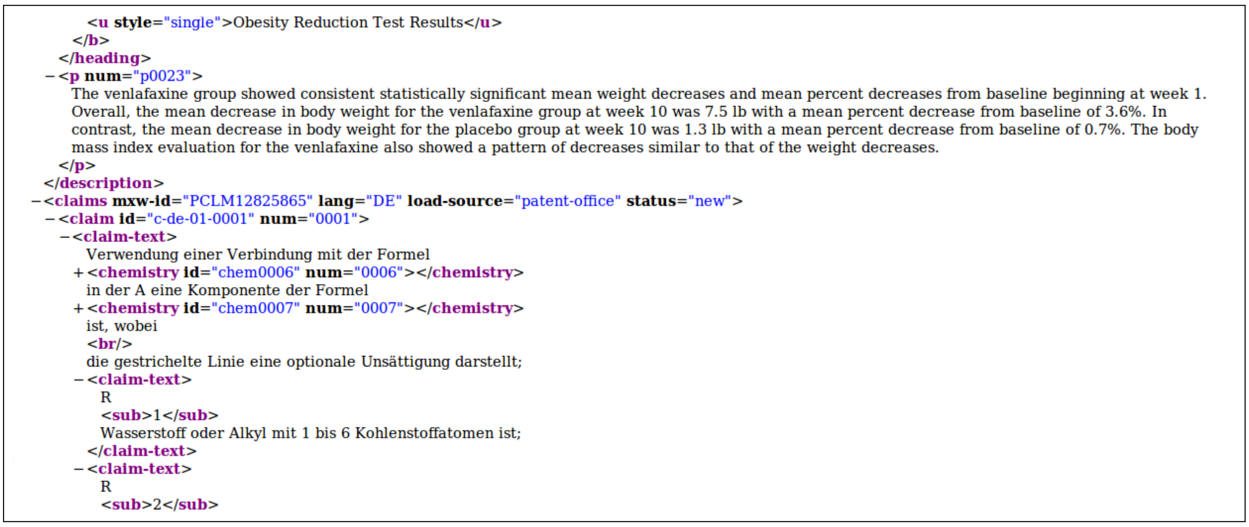
- **A** – Human Necessities
- **B** – Performing Operations, Transporting
- **C** – Chemistry, Metallurgy
- **D** – Textiles, Paper
- **E** – Fixed Constructions
- **F** – Mechanical Engineering, Lighting, Heating, Weapons, Blasting
- **G** – Physics
- **H** – Electricity

The *textual elements* of a patent are the abstract, the description and the claims. Despite most of the patent documents contain abstracts, usually they do not provide descriptions[2]. Each of the three sections has a different purpose: the abstract gives the most relevant information of the invention, the description gives background information for understanding the invention and the series of claims constitute the legal scope of protection of the patent.

A claim is a single (possibly very long) sentence composed mainly of two parts: an introductory phrase and the body of the claim. As it has been said, it is in the body of the claim where there is the specific legal description of the exact invention. Therefore,

[1] International Patent Classification, `http://www.wipo.int/classifications/ipc/en/`

[2] Data source: `http://www.ir-facility.org/prototypes/marec/statistics`



```
        <u style="single">Obesity Reduction Test Results</u>
      </b>
    </heading>
  - <p num="p0023">
      The venlafaxine group showed consistent statistically significant mean weight decreases and mean percent decreases from baseline beginning at week 1.
      Overall, the mean decrease in body weight for the venlafaxine group at week 10 was 7.5 lb with a mean percent decrease from baseline of 3.6%. In
      contrast, the mean decrease in body weight for the placebo group at week 10 was 1.3 lb with a mean percent decrease from baseline of 0.7%. The body
      mass index evaluation for the venlafaxine also showed a pattern of decreases similar to that of the weight decreases.
    </p>
  </description>
- <claims mxw-id="PCLM12825865" lang="DE" load-source="patent-office" status="new">
  - <claim id="c-de-01-0001" num="0001">
    - <claim-text>
        Verwendung einer Verbindung mit der Formel
      + <chemistry id="chem0006" num="0006"></chemistry>
        in der A eine Komponente der Formel
      + <chemistry id="chem0007" num="0007"></chemistry>
        ist, wobei
        <br/>
        die gestrichelte Linie eine optionale Unsättigung darstellt;
      - <claim-text>
          R
          <sub>1</sub>
          Wasserstoff oder Alkyl mit 1 bis 6 Kohlenstoffatomen ist;
        </claim-text>
      - <claim-text>
          R
          <sub>2</sub>
```

Figure 1: An extract of the description and claims sections of a patent document.

claims are written in a lawyerish style and use a very specific vocabulary of the domain of the patent.

An excerpt of a patent description and the text of the patent claim can be seen in Figure 1. At the top of the figure there is a fragment of the description written in English. Next, the series of claims in the available languages are listed. Each claim consists of a sequence of texts which may contain figures, formulae or, as in this case, chemistry items.

The results of this workpackage will be tested by translating abstracts and claims of patents of a given IPC code. In particular, we restrict our testing domain to IPC A61P which corresponds to:

A – Human Necessities

- **A61** Medical or veterinary science; hygiene
  - **A61P** Specific therapeutic activity of chemical compounds or medical preparations

Besides the generalities of patents, this domain uses specific vocabulary of chemistry plenty of names of compounds, which have a particular structure that cannot be properly analysed with standard NLP tools. Therefore, as a collateral effect of the domain, some basic linguistic processors have been built and are summarised in Section 2.3 and further explained in Deliverable 5.2 [12]. Before this, the corpora of patents used are introduced in the following section.

2.2 Corpora

As an European project that aims to translate among the European languages, MOLTO works with European patents. The European Patent Office[3] (EPO) is then a natural provider for the data. The agency registers all the patent entries at least in their three official languages (European Patent Convention, Art.14) and therefore, our task is restricted to these three languages: English, French and German.

Despite the original language in which a patent is written, the specifications of the European patents are published in one of the official languages, and shall include a translation of the claims in the other two official languages. In consequence, the EPO is creating three parallel corpora of claims in the three official languages and several bilingual corpora made up of descriptions in the European languages [42].

At the beginning of the project EPO data was not used though. Matrixware started as a partner of the project, so the corpora at their disposal (MAREC) was used instead. Their corpora included not only European patents but also American and Japanese ones. Due to the nature of the project, only European patents with manual translations in the languages enumerated were considered.

2.2.1 MAREC corpus

MAREC is a data collection over 19 million of European, US and Japanese patent applications and granted patents from 1976 to June 2008. In MAREC, the majority of the documents are written in English, German and French, and about half of the documents include full text. The documents follow a unified XML format normalised from sources of the European Patent Office, World Intellectual Property Organisation, United States Patent and Trademark Office and Japanese Patent Office (only applications). The standardised fields include dates, countries, languages, references, person names, and companies as well as rich subject classifications. Some figures and descriptions of this corpus and the following ones can be found in Deliverable 5.1 [11].

A part of MAREC is made available by the CLEF Initiative (Conference and Labs of the Evaluation Forum[4]) to those scientifics who participate in their Information Retrieval competitions. In particular, we gathered a parallel corpus in the three languages from the corpus of patents given for the CLEF-IP track in the CLEF 2010 Conference[5]. These data are an extract of the MAREC corpus which contain over 2.6 million patent documents pertaining to 1.3 million patents from the EPO with some content in English, German and French, and extended by documents from the WIPO[6].

The corpus provided for the CLEF-IP tasks contains the bibliographic data, abstract, description, and claims. However, not all documents have content in all the fields, and in

[3] http://www.epo.org/

[4] http://www.clef-initiative.eu/

[5] http://clef2010.org/

[6] World Intellectual Property Organization, http://www.wipo.int

SET	Fragments	EN tok	DE tok	FR tok
Training	279,282	7,954,491	7,346,319	8,906,379
Development	993	29,253	26,796	33,825
Test	1,008	31,239	28,225	35,263

Table 1: Statistics for the patents parallel corpus on the biomedical domain in English (EN), German (DE) and French (FR).

MOLTO one needs those patents with translated abstracts and claims. The text of these fields is divided into several fragments, which are well marked. This is important since one needs the text aligned in the several languages to build the parallel corpus.

The complete corpus has 2,680,604 patent documents, 822,144 of which are granted patents. 510,183 out of these have the claims translated into the three languages. There are 119,337 documents with IPC code A61P, the one we chose to represent the biomedical domain. Within this group, 21,150 granted patents have claims in the three languages which are aligned and are used to build the parallel corpus.

One can see in Figure 1 the structure of claims in the XML document. A claim is, in general, a long sentence splitted in fragments marked with <claim-text> tags, probably with nested elements. We search in every patent with trilingual claims, and count the number of *claim-text* elements. Whenever its number is the same for the three languages we assume that the claim is aligned and we add the aligned fragments to the corpus. So, our minimum aligned unit is shorter than a claim and, consequently, shorter than a sentence.

Even though fragments are shorter than a sentence, they may have a large number of words. For an appropriate use of the standard SMT software (`GIZA++` [32] and `Moses` [20]), the final corpus contains only those fragments with less than 100 tokens and with a ratio between the lengths of the source and target sentence less than 9. This methodology leads to 281,283 aligned parallel fragments as it can be seen in Table 1.

Besides, each of the fragments is cleaned in order to achieve an homogeneous corpus. Tags such as <sub> or <br> are removed, chemistry formulae and images with the appropriate tag are substituted by ***IMAGE*** and extra spaces are removed for example.

The final parallel corpus is splitted in three parts. The largest part corresponds to the training corpus and has a total of 279,282 fragments and around 8 million tokens depending on the language (see Table 1). For every language, its side of the parallel corpus is used as a monolingual corpus to estimate the language model in the translation process. Two smaller sets have been selected for development and test purposes, keeping 993 fragments for development and 1008 for test.

2.2.2 EPO facilitated corpus

EPO has provided the project with two different datasets. On the one hand, we have obtained a parallel corpus composed of 1.7M fragments aligned by language pairs. This corpus only contains the parallel raw text and the identifier of the patent, loosing all the

SET	DE-EN	FR-EN	FR-DE
Training	279,282	279,282	279,282
Development	993	993	993
Test MAREC	1,008	1,008	1,008
Test EPO_{MT}	847	858	831

Table 2: Number of aligned fragments used for training and testing the translation systems.

richness of the original XML files, which, on the other hand, is not needed to train a translation system. However, less than 900 fragments from 66 patents (847 for German-English, 858 for French-English and 831 for French-German) correspond to the domain we tackle having IPC code A61P. Therefore, this small dataset, EPO_{MT}, is going to be used for test purposes (see Table 2 for a comparison between the data sets used).

On the other hand, EPO provided also a website from where we downloaded 7,705 patent documents, also in the biomedical domain, all dated from 2010 to 2012. The patent documents follow the normalised XML format defined by the EPO and described before and therefore they are processed in a similar way as in Section 2.2.1. Within this downloaded corpus, up to 4,274 out of the 7,705 documents have claims, and 2,058 out of them are trilingual. 2,116 documents have claims written only in English, 66 have claims only in German, 34 only in French.

These documents are not be used for training the SMT systems. Instead, they are used to populate the patents retrieval system as explained in Section 8 and translated using the engines here described. Notice that our training corpus has patents up to 2008 and this dataset starts from 2010, so there is no overlapping patents and all translations are fair.

2.3 Linguistic processors

The detection and correct tokenisation of chemical compounds has been shown to be crucial in the performance of translators (see Section 3 for the analysis). A regular tokeniser would for example split the compound:

cis-4-cyano-4-(3-(cylopentyloxy)-4-methoxyphenyl)cyclohexane-1-carboxylic

into 9 tokens

cis-4-cyano-4-, (, 3-, (, cylopentyloxy,), -4-methoxyphenyl,), cyclohexane-1-carboxylic,

when using standard tokenisation rules. Consequently, each of the tokens would be translated as an independent word. To deal with this peculiarity of the domain, we developed a pipeline to detect, tokenise and translate compounds. A general scheme can be seen in Figure 2.

We devise a recogniser and tokeniser based on affix detection. A list with approximately 150 affixes has been compiled (142 elements for English and German, and 148 for French)

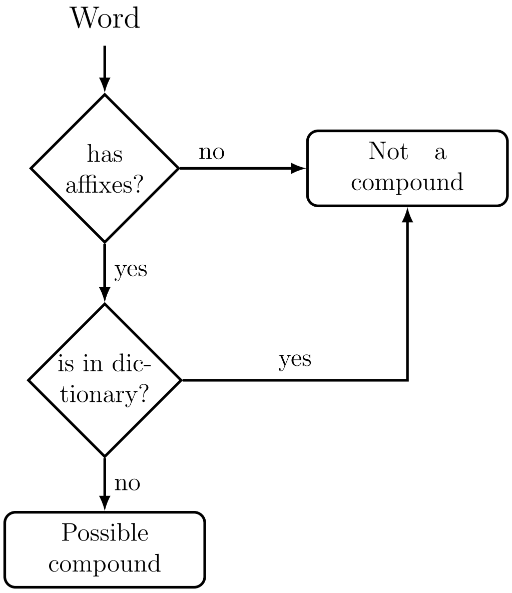


Figure 2: Pipeline to detect candidates to be a chemical compound. Those candidates should not be tokenised.

and it is used to select the candidate tokens to be a compound from the corpus. The list includes prefixes such as *Meth-, Eth-, Prop-, Pentadec-, imido-, selenocarboxy-, hydroxy-, Propion-, Arachid-...* and suffixes such as *-ol, -one, -al, -aldehyde, -oic, -oate, -oxy, -sulfonic, -nitrile, -amine* or *-isocyanide.*

The candidates selected this way are matched against a dictionary and those without a match are considered to be compounds and do not get an internal tokenisation. 103,272 compounds are found with this procedure within the training corpus defined in Section 2.2.1.

However, this list of compounds contains some noise. Examples of noise are in this context proper names with the defined affixes (*Hôpital*), words that do not appear in the dictionary (*extracorporeal*) or simply typos (*comparoate*). The amount of noise is considerable, but extra words do not in general imply a wrong tokenisation. So, the method works better as a (non-)tokeniser than as a compound detector and it bets for high recall instead of precision. Notice also that multiword compounds such as *Potassium bromide* cannot be detected with this methodology, but again, there is no negative effect on tokenisation. This pre-process is applied to the parallel corpus before training the SMT system.

3 Statistical Machine Translation System

3.1 Translation System

In Statistical Machine Translation (SMT) and within the log-linear model [31], the best translation $\hat{e}$ for a given source sentence f is the most probable one, and the probability is expressed as a weighted sum of different elements:

$$T(f) = \hat{e} = \mathrm{argmax_e} \sum_{m} \lambda_m h_m(f, e)\,. \tag{1}$$

In the standard most simple form, one considers 8 components being $h_m(f, e)$ log-probabilities of: the language model $P(e)$, the generative and discriminative lexical translation probabilities $lex(f|e)$ and $lex(e|f)$ respectively, the generative and discriminative translation models $P(f|e)$ and $P(e|f)$, the distortion model $P_d(e, f)$, and the phrase and word penalties, $ph(e)$ and $w(e)$.

The λ weights, which account for the relative importance of each feature in the log-linear probabilistic model, are commonly estimated by optimising the translation performance on a development set. For this optimisation one can use Minimum Error Rate Training (MERT) [30] where BLEU [33] is the reference score.

In our experiments, we build a state-of-the-art phrase-based SMT system trained on the biomedical domain with the corpus described in Section 2.2. Its development has been done using standard freely available software. A 5-gram language model is estimated using interpolated Kneser-Ney discounting with `SRILM` [41]. Word alignment is done with `GIZA++` [32] and both phrase extraction and decoding are done with the `Moses` package [21, 20]. The optimisation of the weights of the model is trained with MERT against the BLEU evaluation metric. Our model considers the language model, direct and inverse phrase probabilities, direct and inverse lexical probabilities, phrase and word penalties, and a non-lexicalised reordering.

As a byproduct of training the SMT system lexical and phrase tables are obtained. The word-to-word translations are deduced from the alignments given by `GIZA++`. Also from these alignments and using the heuristic "grow-diag-final" and the phrase extraction script in `Moses` we obtained the final translation tables used for decoding. These resources are also important to build the patent's lexicon that uses the GF engine (see Section 4).

3.2 Automatic evaluation

Manual evaluation is the most reliable way to quantify the quality of a translation but it is also a very costly way (both in time and money). For a fast and objective evaluation during the system development one needs to make use of automatic metrics. Automatic metrics are costless, objective and reusable but they cannot capture all the aspects that a human evaluator takes into account. In order to somehow overcome this limitation we do not use a single metric such as the standard BLEU but an heterogeneous set.

The `Asiya` evaluation package [15] includes more than 500 metrics and their variants at lexical, syntactic and semantic levels. Syntactic and semantic metrics need linguistic processors to annotate the translations so, the number of available metrics depends on the availability of these processors in the language to be applied. Lexical metrics are in general available to all the languages. In order to be applied to biomedical patents, `Asiya` has been modified including new functionalities and adapting the linguistic processors for English, French and German (see Deliverable 7.3 [26] for further information).

In the following we select a subset of lexical and syntactic metrics to be applied to English, French and German; and a larger set including semantic metrics to be applied to the English translations. The sets are build up with:

Lexical metrics

- PER [43], TER [39], WER [29]: Based on edit distances
- BLEU, NIST [9], ROUGE [22]: Based on n-gram matching (lexical precision: BLEU, NIST; and lexical recall: ROUGE)
- GTM [27], METEOR [5]: Based on the F-measure
- ULC [17]: *U*niform *L*inear *C*ombination. When applied to lexical metrics it includes WER, PER, TER, BLEU, NIST, ROUGE-S*, GTM-2, METEOR-st.

Syntactic metrics [16]

- SP-O_{p}(*), SP-pNIST-5: Based on the lexical overlap or the NIST score over the *p*art-of-speech (Shallow Parsing)
- CP-O_{p}(*), CP-O_{c}(*), CP-STM-9: Based on the lexical overlap among *p*art-of-speech or *c*onstituents of constituent parse trees. Also the *S*yntactic *T*ree *M*atching is included [24] (Constituent Parsing)

In all cases results are given as the average over all types of the corresponding elements.

Semantic metrics [16]

- SR-O_{r}, SR-O_{r}(*): Based on the lexical overlap between semantic *r*oles, or independently of their lexical realization (Semantic Roles)
- DR-O_{r}, DR-O_{rp}, DR-STM-9: Based on lexical overlap or morphosyntactic overlap between discourse *r*epresentations structures (i.e., between *p*arts-of-speech associated to lexical items) or STM metric over these elements (Discourse Representations)

As before, results are given as the average over all types of the corresponding elements.

The uniform linear combination of metrics used in the evaluation at the three linguistic levels considers the metrics in the lexical ULC described above and all the syntactic and semantic here introduced.

	DE2EN			FR2EN		
METRIC	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**
WER	47.07	29.04	**26.34**	46.90	27.30	**21.51**
PER	30.32	16.96	**16.74**	29.37	15.97	**12.34**
TER	43.03	27.02	**24.35**	42.88	25.09	**19.58**
BLEU	44.05	**65.79**	65.41	43.70	67.97	**70.45**
NIST	8.04	**10.46**	10.12	8.28	10.65	**10.86**
GTM-2	31.74	**48.38**	47.94	31.45	50.54	**54.12**
MTR-st	35.92	**46.73**	45.97	38.01	48.29	**49.63**
RG-S*	54.39	**74.75**	71.61	58.53	76.61	**77.82**
CP-Oc(*)	50.60	**58.32**	57.59	39.91	58.93	**63.66**
CP-Op(*)	57.31	64.07	**64.89**	49.16	62.85	**67.38**
CP-STM-9	36.23	36.88	**39.90**	23.73	39.88	**47.31**
SP-Op(*)	54.26	**70.79**	69.89	56.34	72.47	**75.16**
SP-pNIST-5	6.51	8.28	**8.36**	6.49	8.49	**9.01**
SR-Or	18.31	**32.47**	32.42	24.71	39.68	**43.53**
SR-Ori	9.58	**24.55**	24.12	14.96	32.06	**35.53**
DR-Or	28.25	41.98	**50.20**	29.78	56.99	**60.41**
DR-Orp	39.32	50.32	**57.07**	40.29	62.50	**65.94**
DR-STM-9	33.09	46.04	**50.89**	35.05	60.71	**62.34**
ULC	59.17	87.34	**89.85**	52.58	85.62	**92.58**

Table 3: Automatic evaluation using an heterogeneous set of metrics of the in-domain SMT system for translations into English for the MAREC test set. Results of two state-of-the-art systems, Bing and Google, are showed for comparison.

3.3 Patent SMT performance

The evaluation of the statistical system is done on the MAREC test set and on EPO$_{\text{MT}}$. We show the evaluation on the six directions of translation between English, French and German. Translations into English can be deeper analysed because more metrics are available; for translations into French and German we are restricted to the lexical and syntactic metrics.

Together with our system labelled as MOLTO$_{\text{SMT}}$, we show the same evaluation for two public SMT systems for general translation: Bing[7] and Google[8]. These systems can be considered the state-of-the-art of an SMT open domain translator.

[7] http://www.microsofttranslator.com

[8] http://translate.google.com

	EN2DE			EN2FR		
METRIC	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**
WER	53.94	40.20	**30.93**	39.78	32.46	**27.16**
PER	40.49	26.99	**22.82**	28.14	20.82	**18.00**
TER	51.71	38.29	**29.33**	37.70	30.54	**25.57**
BLEU	36.30	53.43	**57.59**	45.73	57.86	**62.40**
NIST	6.95	9.01	**9.40**	8.64	9.70	**9.95**
GTM-2	27.95	39.89	**42.98**	32.67	40.28	**44.99**
MTR-st	51.27	63.34	**67.84**	61.55	71.56	**75.71**
RG-S*	42.74	62.35	**63.14**	58.50	71.40	**72.94**
CP-Oc(*)	35.79	49.88	**52.89**	49.65	60.17	**63.71**
CP-Op(*)	34.87	47.23	**51.62**	51.93	62.56	**65.91**
CP-STM-9	20.49	29.71	**32.99**	36.95	46.75	**50.92**
SP-Op(*)	59.54	65.23	**69.06**	40.34	45.48	**47.54**
SP-pNIST-5	5.47	6.23	**6.95**	3.27	3.46	**3.60**
ULC	54.42	78.32	**86.89**	61.49	77.58	**84.61**

Table 4: Automatic evaluation using a set of lexical metrics of the in-domain SMT system for translations from English for the MAREC test set.

	DE2FR			FR2DE		
METRIC	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**
WER	56.26	37.30	**34.36**	69.29	43.75	**35.02**
PER	37.00	24.49	**22.70**	53.80	29.18	**25.75**
TER	52.53	34.88	**32.29**	66.95	41.79	**33.55**
BLEU	32.14	52.19	**56.02**	24.51	48.75	**52.86**
NIST	6.96	**9.17**	9.09	5.57	8.56	**8.88**
GTM-2	24.88	37.28	**40.93**	22.12	37.19	**40.63**
MTR-st	48.74	66.85	**68.82**	42.37	60.07	**63.00**
RG-S*	43.63	**65.66**	64.03	31.08	56.76	**57.33**
CP-Oc(*)	39.14	56.13	**57.45**	28.33	46.73	**49.70**
CP-Op(*)	42.18	58.98	**60.00**	28.11	44.16	**48.17**
CP-STM-9	27.90	42.41	**44.29**	16.75	27.12	**30.67**
SP-Op(*)	45.85	**64.43**	64.33	28.11	44.16	**48.17**
SP-pNIST-5	5.43	6.88	**6.95**	4.58	6.01	**6.58**
ULC	52.47	82.56	**85.59**	44.85	80.91	**88.58**

Table 5: As in Table 4 for the French-German language pair.

	DE2EN			FR2EN		
METRIC	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**	**Bing**	**Google**	**MOLTO$_{\text{SMT}}$**
WER	47.50	32.16	**29.35**	45.65	27.90	**26.73**
PER	29.29	21.55	**18.64**	29.79	18.98	**15.62**
TER	42.96	29.49	**26.57**	42.48	25.97	**24.67**
BLEU	47.21	64.99	**66.38**	45.54	**68.34**	65.12
NIST	8.47	**10.57**	10.38	8.54	**10.77**	10.54
GTM-2	27.19	41.82	**41.88**	28.46	44.33	**44.71**
MTR-st	34.78	**45.37**	44.96	37.71	**47.43**	46.37
RG-S*	53.70	**74.97**	70.60	58.02	**76.97**	74.31
CP-Oc(*)	43.26	**60.95**	58.17	45.63	64.16	**64.40**
CP-Op(*)	50.39	**67.16**	65.81	53.59	69.11	**69.41**
CP-STM-9	28.04	**41.93**	41.39	30.09	46.90	**48.60**
SP-Op(*)	52.48	**71.13**	68.80	55.93	**72.89**	72.09
SP-pNIST-5	6.57	7.96	**8.24**	6.86	8.54	**8.82**
SR-Or	22.71	32.85	**34.83**	27.90	42.28	**42.34**
SR-Ori	12.61	26.10	**26.38**	17.86	**33.88**	33.81
DR-Or	33.54	29.45	**50.24**	32.40	49.24	**57.02**
DR-Orp	43.54	38.78	**55.82**	44.92	57.08	**63.38**
DR-STM-9	35.19	33.20	**50.41**	38.03	52.69	**59.66**
ULC	58.80	81.76	**88.50**	57.72	87.24	**89.83**

Table 6: As Table 3 for the test set EPO$_{\text{MT}}$.

In general, our in-domain trained system performs significantly better than the two general purpose ones. The linear combination of metrics, ULC, is better for MOLTO$_{\text{SMT}}$ in all the cases. This is mainly because of two reasons. First, it has been trained on the specific domain and second, the tokenisation tools have been specifically developed to deal with chemical compounds. The concrete values can be read in Tables 3, 4, 5, 6, 7 and 8. The first three correspond to the MAREC test set and the last three to EPO$_{\text{MT}}$ set.

For some specific metrics and language pairs, Google is better than MOLTO$_{\text{SMT}}$ whereas Bing is always below the other two. The best MOLTO system is that for French to German translation; the worse system involves the translation from German to English. According exclusively to lexical metrics our system shows weaknesses when translating from German: edit distance metrics favour MOLTO$_{\text{SMT}}$ but n-gram matching measures are mainly preferring Google.

The translation from English is a strong point from MOLTO both lexically and syntactically for our in-domain test set. It is only for the German to English translation that Google outperforms our system by comparing its syntax. Semantically, MOLTO$_{\text{SMT}}$ is clearly the best engine, although it must be noted that semantic metrics are only available for English.

	EN2DE			EN2FR		
METRIC	**Bing**	**Google**	**MOLTO$_{\mathbf{SMT}}$**	**Bing**	**Google**	**MOLTO$_{\mathbf{SMT}}$**
WER	53.24	38.06	**35.32**	42.44	30.26	**30.08**
PER	38.46	25.71	**25.59**	30.74	20.76	**19.21**
TER	50.78	35.87	**33.47**	40.01	28.50	**28.35**
BLEU	42.34	57.63	**58.46**	44.21	**61.43**	61.08
NIST	7.62	**9.77**	9.50	8.59	**10.28**	9.94
GTM-2	24.96	37.35	**37.86**	28.02	38.44	**40.75**
MTR-st	50.03	**63.95**	63.44	57.75	71.15	**72.27**
RG-S*	42.40	**64.01**	61.12	57.44	**70.61**	69.90
CP-Oc(*)	35.85	**52.68**	52.07	48.54	60.51	**62.90**
CP-Op(*)	34.65	49.39	**50.15**	52.11	65.11	**66.55**
CP-STM-9	21.06	31.75	**32.75**	37.43	49.74	**52.95**
SP-Op(*)	60.16	67.12	**68.17**	60.96	73.37	**73.64**
SP-pNIST-5	5.45	6.38	**6.64**	7.02	**8.04**	7.99
ULC	56.35	82.94	**83.99**	60.14	82.33	**83.87**

Table 7: As Table 4 for the test set EPO$_{\mathrm{MT}}$.

	DE2FR			FR2DE		
METRIC	**Bing**	**Google**	**MOLTO$_{\mathbf{SMT}}$**	**Bing**	**Google**	**MOLTO$_{\mathbf{SMT}}$**
WER	57.84	**39.07**	39.36	67.51	42.63	**38.54**
PER	40.62	25.25	**24.24**	50.44	28.88	**27.33**
TER	54.33	**36.07**	36.50	64.85	40.12	**36.49**
BLEU	30.30	52.47	**53.69**	26.46	**52.07**	50.64
NIST	6.85	**9.27**	8.96	5.89	**8.90**	8.72
GTM-2	21.79	33.67	**35.92**	19.62	34.47	**35.82**
MTR-st	44.49	64.79	**66.38**	41.58	**60.49**	60.22
RG-S*	41.05	**63.49**	61.00	30.88	**58.01**	55.74
CP-Oc(*)	37.18	54.45	**55.83**	28.14	48.02	**49.36**
CP-Op(*)	41.23	58.56	**60.11**	27.71	45.24	**46.64**
CP-STM-9	28.38	43.57	**45.23**	16.91	27.96	**30.38**
SP-Op(*)	44.40	**63.46**	63.32	27.71	45.24	**46.64**
SP-pNIST-5	5.56	**7.06**	6.96	4.56	5.86	**6.33**
ULC	51.49	83.40	**84.32**	45.94	83.83	**86.41**

Table 8: As in Table 7 for the French-German language pair.

DE	Verwendung nach Anspruch 23 , worin das molare Verhältnis von Arginin zu Ibuprofen 0,60 : 1 beträgt .
EN	**The use** of claim 23 , wherein the molar ratio of arginine to ibuprofen **is** 0.60 : 1 .
Domain	The use of claim 23 , wherein the molar ratio of arginine to ibuprofen 0.60 : 1 .
Google	The **method** of claim 23 , wherein the molar ratio of arginine to ibuprofen 0.60 : 1 **is** .
Bing	Use of claim 23 , wherein the molar ratio of arginine to ibuprofen is 0.60 : 1 .

DE	(±)-N-(3-Aminopropyl)-N,N-dimethyl-2,3-bis(syn-9-tetradecenyloxy)-1-propanaminiumbromid
EN	(±)-N-(3-**a**minopropyl)-N,N-dimethyl-2,3-bis(syn-9-tetradeceneyloxy)-1-propanaminium **bromide**
Domain	(±)-N-(3-Aminopropyl)-N,N-dimethyl-2,3-bis(syn-9-tetradecenyloxy)-1-propanaminiumbromid
Google	(±)-N-(3-aminopropyl)-N , N-dimethyl-2 , 3-bis (syn-9-tetradecenyloxy) is 1-propanaminiumbromid
Bing	(±)-N-(3-Aminopropyl)-N,N-dimethyl-2,3-bis(syn-9-tetradecenyloxy)-1-propanaminiumbromid

Table 9: Examples of wrong German-to-English translations in SMT systems.

Looking at the most common lexical metric, BLEU, MOLTO$_{\text{SMT}}$ is the best system in 8 out of 12 evaluations. However, when all aspects are combined, that is using ULC the number increases to 12 out of 12. Still, in some aspects Google and MOLTO$_{\text{SMT}}$ are even.

Notice that the evaluation shown here differs from that reported in Deliverable 5.2. Translations from Bing and Google have been done again with updated versions of the translators. The first evaluation used the systems as they were in February 2011, whereas the current one uses the systems as performed in April/May 2012. Google signed an agreement with the EPO in March 2011 so, their training corpus might already contain EPO's patents at the time we have done the second evaluation. This is an explanation for the large improvement of Google's translator on our test sets during the last year. In the case of Bing, results are slightly different but without any significant improvement.

Even though the MOLTO$_{\text{SMT}}$ system shows a good performance among SMT systems, some of the observed translation errors would not be produced by a rule-based system, which, on the other hand, would probably produce different ones. Table 9 displays two translations from German into English where this is made evident. In the first one, systems are not able to capture the different order in the verb position, although the translation is adequate lexically. The second sentence is an example of the importance of the chemical names. Google, for instance, tokenises the compound by the punctuation. Some of the tokens are then translated, but the full compound is not recovered. Bing and MOLTO$_{\text{SMT}}$ do not tokenise the compound, but according to the results, the word does not appear in the training corpus and has not been translated. Ordering or concordance errors could be easily alleviated by a GF grammar and are a motivation to combine GF and SMT for the translation of patents.

4 Grammatical Framework System

GF is a type-theoretical grammar formalism, mainly used for multilingual natural language applications. Grammars in GF are represented as a pair of an *abstract syntax* –an interlingua that captures the semantics of the grammar on a language-independent level, and a number of *concrete syntaxes* –representing target languages. There are also two main operations defined, *parsing* text to an abstract syntax tree and *linearising* trees into raw text.

The GF resource library [35] is the most comprehensive grammar for dealing with natural languages, as it features an abstract syntax which implements the basic syntactic operations such as predication and complementation, and 20 concrete syntax grammars corresponding to natural languages. This layered representation makes it possible to think of multilingual GF grammars as a RBMT system, where translation is possible between any pair of languages for which a concrete syntax exists. However, the translation system thus defined is first limited by the fixed lexicon defined in the grammar, and secondly by the syntactic constructions that it covers. For this reason, GF grammars have a difficult task in parsing free text. Current work on patent translation has been the first attempt to use GF for parsing un-annotated free text [10]. On the other hand, there is some recent work on parsing the Penn Treebank with the GF resource grammar for English [2], this robust parsing is also being applied to the translation of patents in a second type of systems.

The extension of GF to a new domain implies the construction of a specialised grammar that expands the general resource grammar. Since in our case of application we are far from a close and limited domain, some probabilistic components are also necessary.

Besides, because of the length of the sentences and the complicated grammatical structures, a GF grammar-based system alone cannot parse most of full sentences. This is the problem about parsing free-text we have just said. Consequently, a first family of systems aims at using GF for translating patent chunks, and assemble the results in a later phase. These chunks are obtained by using external processors to GF.

Most of the work and experiments have been done for the French grammar, whereas the German one has been done in his likeness and improved afterwards with the particularities of German.

4.1 French Grammar

The overall system functioning was already introduced in Deliverable 5.2 so here we summarise the system architecture and introduce the new modules and modifications to the main architecture.

4.1.1 General architecture

Figures 3 and 4 show the main modules of the GF translator. All the necessary preprocess before really translating a chunk is depicted in the first figure. After tokenising the text with the processor of Section 2.3, claims are tagged with part-of-speech (PoS) with `Genia` [45],

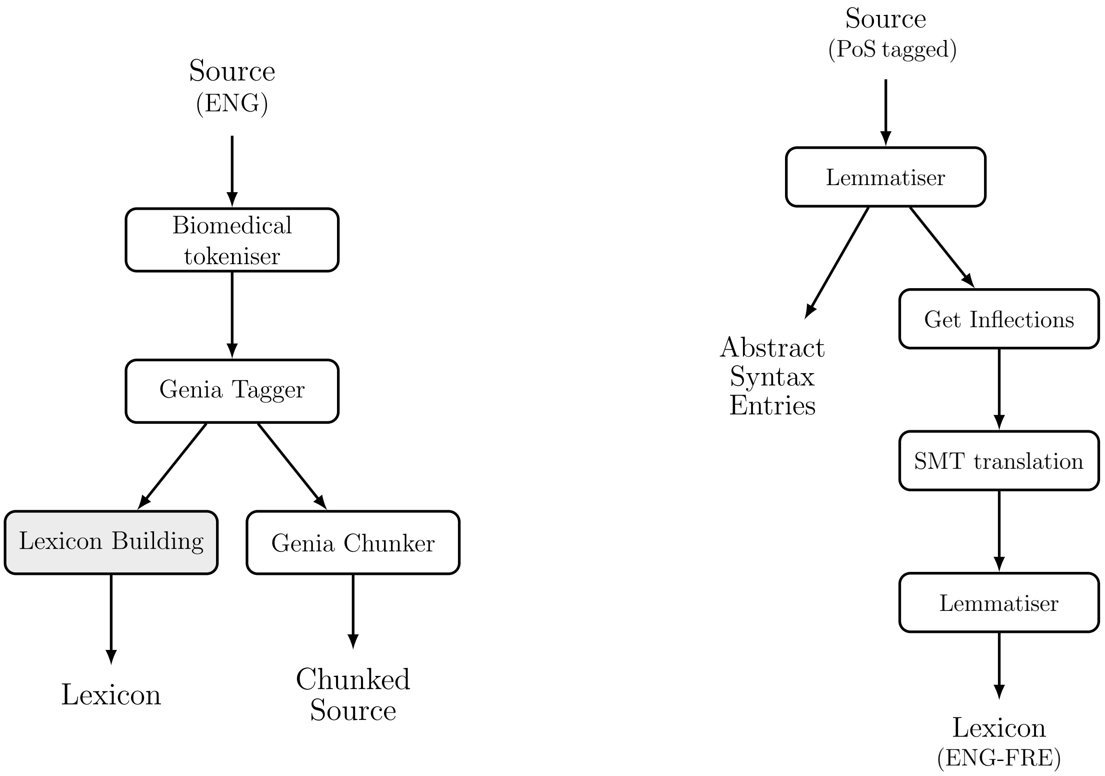


Figure 3: First modules of the GF translator. The input is preprocessed and used to build the lexicon and mark the chunks to translate. The rhs details the lexicon building process.

a PoS tagger trained on the biomedical domain. The output of the tokeniser is used for two purposes, as input for the chunker and as the source to build the lexicon.

From the PoS-tagged words only the ones labelled as nouns, adjectives, verbs and adverbs are kept, since the GF library already has an extensive list of functional part-of-speech such as prepositions and conjunctions. The extensive GF English lexicon[9] is used as a lemmatiser for the PoS-tagged words, so that one can build their correspondent abstract syntax entry (rhs of Figure 3). Moreover, all the inflection forms of a given word are obtained from the same resource. This process is made online. For every sentence to translate, the lexicon is enlarged with the corresponding vocabulary. The French version of the lexicon is built by translating the individual entries from the English lexicon (all inflection forms) with the SMT individual system trained on the patent corpus. The French translations are lemmatised with an extensive GF French lexicon, based on the large morphological lexicon Morphalou [37] in order to get their inflection table. The

[9] The GF English lexicon is based on the Oxford Advanced Learner's Dictionary, and contains around 50,000 English words.

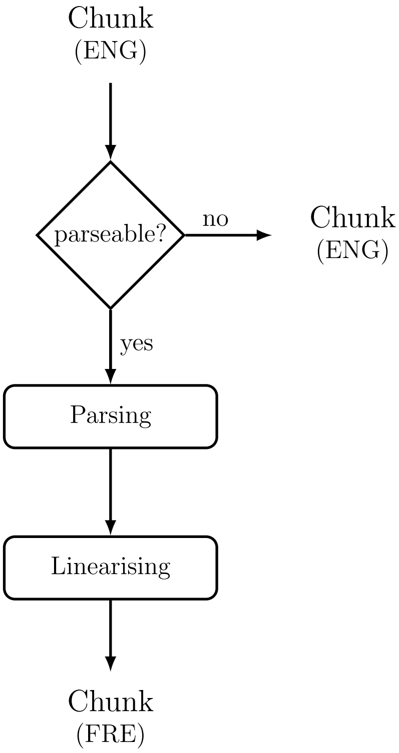


Figure 4: Chunk translation. Those that cannot be translated by GF are kept in English and translated by the SMT system.

part-of-speech is assumed to be the same as in the English counterpart.

The following step is chunk translation for what the grammar has been adapted (Figure 4). The source text is chunked into noun phrases (NP), adjective phrases (AdjP), adverbial and prepositional phrases (AdvP), relative pronouns (RelP) and verb phrases (VP). Other kinds are ignored. Whereas NPs can be translated directly, a VP, RelP or AdjP needs to have an NP to agree with, otherwise the GF grammar cannot know which linearisation form to choose. For NP and AdvP which can be translated independently, a mapping into corresponding GF categories is defined, whereas for VP, RelP and AdjP, their GF mapping requires an NP in order to build their correspondent linearisation. If the required NP is not found, the chunk is sent to the SMT. Also, the VP category from the English and French GF resource grammars is implemented as a discontinuous category, so that it can handle discontinuous constituents in English and clitics in French. The patent grammar uses a category built on top of VP, which represents the flattened version of a VP, with all the constituents combined.

After the annotation, a postprocess is necessary to deal with the specifics of patents. Some PoS tags are updated and merged. For instance, the token “said” is tagged as a verb whereas in patents it acts as a definite article (*The use of claim 1, wherein* said *use is* *intramuscular.*). With the wrong tagging it would not be merged with the following noun into a noun phrase. That sometimes forces to modify the PoS tag of the following word and changes some of the chunks labels.

The last step is the translation of the individual chunks. The translation sequence is done from left to right, so that the last-occurring NP is retained, and is used to make the agreement with VP, RelP or AdjP. If no such NP can be found, or if the GF grammar is not capable to parse the one indicated by the chunker, the current chunk is passed to the SMT. If the grammar can parse, it parses the corresponding English chunks to obtain a forest of abstract syntax trees. In order to disambiguate among the possible options, all of them are linearised, looked up in the French corpus and the most frequent linearisation is kept as the best translation. For the system where we use more than one GF translation this step is ignored. Finally, chunks are combined together with the punctuation marks, other non-included elements and untranslated chunks in the same order as in the source language.

We call this system Version 1. Together with a full hybrid system that makes use of it, it has been published in Ref. [10] and explained in the previous deliverable. After these first results, the system was modified in order to make all the components interact automatically as a one-click system, avoiding the need of any manual intervention. The details of the final architecture are given in the following.

4.1.2 Structure

As it has been said before, the patent grammar is build on the resource grammars with several additions for dealing with chunks. The main contributions are:

- 4 new categories for parsing chunks (`Parse_AdjP, Parse_AdvP, Parse_VP, Parse_NP`) and 3 for agreement among them (`NP_VP, NP_AdjP, NP_RelP`)

- 8 functions that deal with these agreements and the parsing of chunks

- 9 extensions to the resource grammar dealing with general-purpose constructions not supported so far. Examples of these constructions with descriptive names are `Nominalisation` ("*recovering* the antibodies"), `Aggregation` (*mouse antibody* – antibody of mouse), `Adjectivisation` ("*immunised* mouse") or `GerundAsAdj` ("*following* states")

- 13 constructions to deal with claim-specific data such as *any of the claims, of claim*, marks of image, chemical compounds or numbers as proper names

- 8 structural words typical for patent domain case such as *which, said, wherein, according to* or *characterized in that*

4.1.3 Domain-specific lexicon

Since the lexicon for the prototype system in Version 1 was acquired semi-automatically, we devised several ways of fully automating the process. For this, one needs to construct a base dictionary from which the final lexicon are built. Two different base dictionaries are created:

Core Lexicon. The core lexicon is made of 198 words. These elements are selected as the most common English words appearing in the training corpus used in the SMT system. The English words have been translated to French and German and, since it is a small dictionary, it has been manually checked assuring a correct translation. The dictionary contains only single words, not multiple mappings or phrases but it represents the starting point for the online lexicon acquisition.

Static Lexicon. The static lexicon is a larger lexicon made of 3,983 words. It is built from the translation tables obtained with `Moses` from the `GIZA++` alignments on the parallel SMT training corpus. This parallel lexicon is constructed by looking for the most likely translation of the English words in the translation tables. Contrary to the core lexicon, the static one contains one-to-many translations. The resource has been also manually checked.

4.2 German Grammar

4.2.1 Structure

In the final version of the translator there is no difference in the basic structure for the grammars for French and German as the grammars are built modularly and they share most of the modules. So, the German patents grammar is built on the resource grammar with the same additions of Section 4.1.2 and some specifications. The only addition exclusively for German is a component for analysing compounds such as *Nucleotidsequenz* (nucleotide+sequence). This implied the development of a small grammar with 10 constructs to deal with a compositional way of forming compounds. This grammar was necessary to build a dictionary for compounds (see below).

Nominalisation. The function for nominalisation has been reimplemented in order to reflect German idioms – for instance, German does not have nominalisation of verbs via the gerund form (which form does not even exist in the language); instead, the infinitive is used. The definite article, despite not being strictly necessary, was also added to match better with the German files – while there are a number of cases where the article is left out and while it can reasonably be omitted in sentence-initial position, if the register used in the text is appropriate (which the patent claims certainly are), the original claims also have occurrences of the definite article, which in many cases makes the text more readable.

For this, the function has to take a VPSlash instead of a fully formed VP, so the infinitive can be extracted.

```
NominalisationSlash vs np = {
  s = \\c => (DefArt).s ! False ! R.Sg ! R.Neutr ! c ++ "|+"
    ++ vs.s.s ! R.VInf False ++ S.possess_Prep.s
    ++ np.s ! R.NPP R.CVonDat ;
    a = np.a ; isPron = False ; hasComma = False
   } ;
```

The |+ symbol indicates that the following word (in this case, the verb infinitve) be uppercased and thus made into a noun. Thus, *immunising the mouse* becomes *das Immunisieren von der Maus*. While it would be slightly more idiomatic to use the genitive case instead of a prepositional construction (*das Immunisieren der Maus*), using a prepositional case is safer, because we have to take mass NPs (e.g. *milk*) into consideration, which, having no article in many cases, behave differently. This is what the function output looks like in the GF interpreter:

```
Patents> l NominalisationSlash
                (SlashV2a contain_V2)
                (DetCN (DetQuant DefArt NumSg) claim_N)
containing the claim
das |+ enthalten vom Anspruch
```

Relative sentence conversion. Another issue that was taken into account is the fact that attributive gerund/participle sentences (such as *pharmaceutical composition comprising an aqueous solution* (...)) are not nearly as common in German as they are in English or even French. While the German patent claims contain their fair share of constructions like *pharmazeutische Zusammensetzung, umfassend*, using a participle construction with auxiliary verbs like *haben* sounds quite awkward. This is why it was decided to replace all participle sentences with the corresponding relative clauses in the chunking steps.

The abstract grammar contains the function `AgreeNP_VPRS : NP → VP → NP_RS ;` which in turn makes use of the conversion function `VPToRS : VP → RS ;`. In the German grammar, the implementation of VPToRS looks as follows, with the new tree being built using parameters from the resource grammar:

```
VPToRS vp = let
  tmp : {s : Str ; a : Anteriority ; m : R.Mood ; t : ParamX.Tense} =
  {s = [] ; a = ParamX.Simul ; m = R.MIndic ; t = ParamX.Pres}
      in
        UseRCl tmp PPos (RelVP Which vp) ;
```

This is being used in the agreement function:

```
AgreeNP_VPRS np vp = let gen : R.Gender = R.genderAgr np.a ;
                         num : R.Number = R.numberAgr np.a ; in
                 {s = (VPToRS vp).s ! (R.gennum gen num) } ;
```

Thus the RP can use the preceding NP's parameters in agreement.

With these constructions, a first system used a grammar with twice as many constructions as the French grammar, but because of the need of making the system uniform in the target languages and the small difference in performance, the German part was modified in order to fit in the initial framework.

4.2.2 Domain-specific lexicon

Core Lexicon. The core lexicon is built with the same 198 most frequent English words as for French.

Static Lexicon. The static lexicon is built using the SMT word alignments. From the aligned word-to-word pairs, the ones with the highest respective probabilities are selected and lemmatised using the GF dictionary and smart paradigms[10]. In the same step, word pairs get matched with a previously SMT-built English lexicon based on the patents corpus; so in the end, we are left with mappings from said English lexicon to entries from the German GF dictionary DictGer.

Using a dictionary extracted from Wiktionary and the word alignment tables, only about 33% of the domain-specific abstract lexicon was covered[11]. With the help of the entries extracted from the patents corpus, coverage could be extended to 50%. After some corrections, the current coverage is now at around 80%, with the remaining entries mostly being PoS mismatches or otherwise faulty. 43,084 entries build up GerLex in this way.

Dictionary of compounds. Aside from cases where word alignment did not succeed (e.g. where verbs would be wrongly mapped to prepositions because of high cooccurrence) or part-of-speech mismatches (which cannot be remedied, at least not automatically[12]), a major problem that had to be tackled while building the lexicon for the patents translation system was the question of how to align compound nouns. While a recogniser and

[10] A smart paradigm is GF function that takes the basic form of a word (singular nominative for nouns, infinitive for verbs, masculine singular for adjectives) and builds the whole GF representation, by using a set of rules for inferring the other forms.

[11] This number does not count proper nouns, which are always translated with SMT and thus need not be included in the GF lexicon.

[12] Consider the pair *carboxylic acid* and *Karboxylsäure*, where the German noun is a compound of two common nouns, and the English one consists of an adjective and a noun. It would be wrong to translate *carboxylic acid* with *karboxylische Säure*, but it is obviously also not possible to map adjectives to nouns.

parser for chemical compounds like *2,6-dimethyl-4-(3-nitrophenyl)-1,4-dihydropyridine-3,5-dicarboxylic acid-3-(N-benzyl-4-piperidinyl)ester-5-methyl ester hydrochloride* already exists, the treatment of German compound nouns is an entirely different and notoriously tricky issue.

The lexicon the GF chunk translator uses is generated using word-to-word alignment with `GIZA++` and contains one-to-one and one-to-many translations. While this was a workable approach in translating English to French, the mappings are less unambiguous in the case of English to German. The English compound *nucleotide sequence* translates to French as *séquence de nucleotides*, which makes it possible to do a mapping from *nucleotide* to *nucleotides* and from *sequence* to *séquence*. However, the German equivalent of *nucleotide sequence* is *Nucleotidsequenz*, as German is a language which compounds nouns by concatenating the constituents. If we now attempt a one-to-one mapping, we will inevitably end up with *nucleotide* → *Nucleotidsequenz* and *sequence* → *Nucleotidsequenz*.

There are two accepted ways to deal with the problem [34]:

- split German compounds in the original corpus used in order to make one-to-one mapping possible (i.e. split *Nucleotidsequenz* into *Nucleotid* and *Sequenz*); rejoin the constituents in a later step

- create new "single-word" compound entries for English (i.e. create a new English word *nucleotide_sequence*, which can be mapped to the German compound)

In our approach we decided to go with the first method by applying a naïve heuristic of looking up substrings of compound words in the dictionary and splitting them if there was a match. With this method a dictionary with 7,774 entries is built.

4.3 System definition and patent GF performance

With the grammar and the lexical resources built for patents one can define different systems. The main differences among these rely on the characteristics of the lexicon and the number of translations used which, in turn, depend on the confidence that GF assigns to them. The following list describes these differences.

Static. Uses the static dictionaries for French and German and does not add new lexical items.

Runtime safe. Uses a base lexicon (core or static) to start with and adds the lexical items that are not present there. The method assumes that the English words are found in the English monolingual dictionary and that the target translation is found in the corresponding target dictionary and has the right PoS tag. We assigned a confidence score to each of the translated chunks. The score is 4 for unambiguous translations and 1 for the ambiguous ones.

Runtime unsafe. Uses a base lexicon (core or static) to start with and adds the lexical items that are not present there. The method adds words regardless whether they are found in the monolingual dictionaries and builds the representation with the help of smart paradigms. This approach is used for nouns, adjectives and adverbs. As before, chunk translations are labelled with confidence scores: 4 for translations that are not ambiguous and contain only "safe" words from dictionaries, 1 for translations that are "safe" but ambiguous, and 0.2 for translations that contain at least an "unsafe" word. Dependent "safe" elements such as RelP, AdvP and VP which depend on an NP are labelled as 0.2 in this system.

Version 1. The first architecture used the runtime unsafe lexicon acquisition without confidence scores. That forced some manual intervention in order to discard wrong translations, most of them actually labelled with a score of 0.2. This option is not used anymore, but it is shown for comparison at some points in the following analysis.

Base. Uses the core lexicon as start lexicon and adds the missing lexical items on top of these by one of the two methods - safe/unsafe.

Extended. Uses the static dictionaries as starting point instead of the core lexicon, and adds the missing lexical items on top of these by one of the two methods - safe/unsafe.

Single. Grammar translations for chunks are one-to-one. In case of ambiguities, and therefore multiple translation options, only the most likely translation is kept.

Multiple. Multiple grammar translations for chunks. In case of ambiguities all the possibilities are kept. Individual translations have their own probability - for the unsafe case where the words that form the chunk play an important role in computing the probability of the chunk translation.

Table 10 summarises these options and the final systems that can be built with them. Notice that neither multiple translations nor confidence scores assigned at this point have any consequence in the translation. The first translation of a chunk irrespective of its *probability* is going to be used for the GF system. However, having multiple options and their confidence is important when combining with other translation options from the SMT system (see Section 7).

Regarding the GF translation a first thing to notice is that the number of total chunks is different from the system introduced in the previous deliverable and in Ref. [10]. In order to be consistent with the translation of the SMT system, the tokeniser used in the GF translation has been changed from that available in `Genia` to our in-house tokeniser. Although both tokenisers are specialised for the biochemical domain results differ and, consequently, the number and type of detected chunks differs as well. The `Genia` chunker with

	Base lexicon	**Lexicon acquisition**	**Multiple GF options**
GF-Sts	–	static	○
GF-SaBs	base	safe	○
GF-UnBs	base	unsafe	○
GF-SaEs	extended	safe	○
GF-UnEs	extended	unsafe	○
GF-Stm	–	static	●
GF-SaBm	base	safe	●
GF-UnBm	base	unsafe	●
GF-SaEm	extended	safe	●
GF-UnEm	extended	unsafe	●

Table 10: Main characteristics and nomenclature for the grammars.

its own tokeniser divides the test set in 11,002 chunks, whereas with the in-house tokeniser the number diminishes to 10,456. The difference is mainly due to over-tokenisation that `Genia` applies to compound names with dashes or commas.

Table 11 shows some figures on the number of chunks that can be translated by GF according to its type. Also the percentage with respect to the total number for every type and the total number of chunks are indicated. Notice that punctuation is not included in these numbers as GF grammars do not deal with it[13]. Considering punctuation would add 3,107 chunks more.

RelPs are the most translated chunks in percentage for any of the systems. That is, words such as *which* or *wherein* are always translated unless there is a lack of agreement. However, they are not the most numerous, and most of the GF contribution comes from NPs and AdvPs. The system with a static lexicon is the one with more difficulties to parse the chunks and this is reflected in a lower percentage of translation for any kind of chunk. The safe and unsafe versions have a similar performance and using the extended base lexicon improves specially the translation of VPs and AdvPs but damages the AdjPs.

There might be several reasons why GF cannot translate the chunks. First and most important, in some cases the chunks cannot be parsed by the GF English grammar. When parsed, there might be a lack of information about agreement or, in case of dynamic lexicons, missing words in the bilingual lexicon. As an example let's see the effect on the GF-Sts system. In this case 6,051 chunks cannot be translated in front of the 4,255 that do. In a 92% of the cases chunks cannot be translated because of the parsing, and the remaining 8% is due to missing the agreement. These numbers are representative of all the systems, with a significant difference when the lexicon is not fully static. For the GF-SaBs system 4,175 chunks are not translated: 84% because the chunk cannot be parsed, 5% because a lack of agreement and the remaining 11% because some words are not found in

[13] This is also true for coordinating conjunctions between chunks.

	Version 1	**GF-Sts**	**GF-SaBs**
NP	2,366 (52.2%)	1,741 (44.35%)	2,781 (70.44%)
VP	275 (5.7%)	489 (30.52%)	702 (43.82%)
AdjP	82 (36.8%)	49 (33.11%)	105 (70.95%)
AdvP	1,960 (50.3%)	1,460 (37.64%)	1,940 (50.01%)
RelP	648 (88.3%)	506 (69.41%)	603 (82.72%)
Sum	5,301 (48.2%)	4,255 (41.29%)	6,131 (59.49%)
Total	11,002	10,306	10,306

	GF-UnBs	**GF-SaEs**	**GF-UnEs**
NP	2,866 (69.94%)	2,811 (71.20%)	2,896 (70.67%)
VP	701 (43.76%)	873 (54.49%)	872 (54.43%)
AdjP	103 (69.59%)	87 (58.78%)	85 (57.43%)
AdvP	1,855 (47.82%)	2,037 (52.51%)	1,952 (50.32%)
RelP	599 (82.17%)	603 (82.72%)	599 (82.17%)
Sum	6,124 (58.57%)	6,411 (62.21%)	6,404 (61.25%)
Total	10,456	10,306	10,456

Table 11: Number of chunks translated by GF and the percentage of the total chunks of the correponding type for the French grammars.

the lexicon.

Comparing Version 1 and the new systems, one can see that the decrease in the number of chunks given by the different tokenisation does not represent a decrease on the total number of chunks that can be translated unless, of course, for the system with only a static lexicon. In fact, the performance has been improved, being the major improvement that of the translation of NP and VP: more than 400 NPs are now translated reaching a 70% of all the parsed NPs. This is probably because the lexicon is better quality after polishing the rules for translating words and this especially applies to nouns since they account for more than half of the lexicon.

To study the quality of the translated chunks we estimate a set of lexical and syntactic metrics on the MAREC and EPO$_{\mathrm{MT}}$ test sets. Since more than half of the chunks are not translated, most of the sentences contain chunks in English, the source language. The coverage is then very low and all the metrics show low scores. The aim of this evaluation is to see which of the systems is performing the best, although all of them will give poor results when considered as a full translation system.

For both test sets and metrics (see Table 12 for MAREC and Table 13 for EPO$_{\mathrm{MT}}$) the best system is GF-SaEs, that is, the GF that uses an online safe lexicon starting with

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-Sts	67.02	57.78	65.74	19.91	4.74	18.24	30.78	21.18	56.20
GF-SaBs	67.96	52.27	65.76	20.41	4.98	18.09	33.64	23.30	60.02
GF-SaEs	**66.62**	**51.14**	**64.47**	**21.69**	**5.17**	**18.87**	**35.36**	**26.20**	**64.45**
GF-UnBs	68.02	52.42	65.84	20.31	4.97	18.04	33.54	23.19	59.75
GF-UnEs	66.68	51.30	64.55	21.57	5.15	18.78	35.25	26.06	64.13

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-Sts	21.55	22.00	17.98	21.65	1.66	84.11
GF-SaBs	25.13	25.95	20.58	24.44	1.91	97.07
GF-SaEs	**25.92**	**26.83**	21.06	**25.21**	**1.97**	**99.98**
GF-UnBs	25.14	25.82	20.63	24.35	1.89	96.70
GF-UnEs	25.89	26.69	**21.09**	25.12	1.94	99.55

Table 12: GF translation for the English-to-French language pair for the MAREC test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-Sts	65.29	56.84	64.29	22.84	5.23	16.36	31.68	20.31	55.34
GF-SaBs	63.26	51.68	61.97	24.35	5.62	16.72	34.69	21.72	60.90
GF-SaEs	**61.92**	**50.35**	**60.48**	**25.36**	**5.81**	**17.42**	**36.26**	**24.97**	**65.31**
GF-UnBs	63.38	51.83	62.09	24.27	5.61	16.66	34.55	21.59	60.60
GF-UnEs	62.03	50.49	60.60	25.26	5.80	17.35	36.12	24.78	64.97

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-Sts	22.21	23.92	19.60	31.19	4.16	86.90
GF-SaBs	24.53	26.32	21.75	34.48	4.49	95.57
GF-SaEs	**25.94**	**28.04**	**22.46**	**36.29**	**4.60**	**100.00**
GF-UnBs	24.36	26.11	21.59	34.34	4.47	94.98
GF-UnEs	25.79	27.87	22.23	36.13	4.58	99.39

Table 13: GF translation for the English-to-French language pair for the EPO_{MT} test set.

the extended one. All the scores are consistent in first preferring those systems that start with the extended lexicon, then those starting from the core lexicon and finally the static version. Again, the differences between the safe and the unsafe acquisition are not so important.

Up to now, all the comments refer to the English-to-French translation but a similar analysis and conclusions can be established for the English-to-German systems. Table 14 shows the number of chunks translated by GF and the percentage of the total chunks of the corresponding type for the German grammars. The number of chunks that can be translated with only a static lexicon are much less than for French: only a 22% of the

	GF-Sts	**GF-SaBs**	**GF-UnBs**	**GF-SaEs**	**GF-UnEs**
NP	1,001 (25.35%)	2,746 (69.55%)	2,779 (67.81%)	2,796 (70.82%)	2,827 (70.67%)
VP	186 (11.61%)	676 (42.20%)	674 (42.07%)	742 (46.32%)	741 (54.43%)
AdjP	21 (14.19%)	106 (71.62%)	104 (70.27%)	97 (65.54%)	97 (57.43%)
AdvP	550 (14.18%)	1,876 (48.36%)	1,843 (47.51%)	1,871 (48.23%)	1,840 (50.32%)
RelP	506 (69.41%)	603 (82.72%)	566 (77.64%)	603 (82.72%)	590 (82.17%)
Sum	2,264 (21.97%)	6,007 (58.29%)	5,966 (57.06%)	6,109 (59.27%)	6,095 (59.29%)
Total	10,306	10,306	10,456	10,306	10,456

Table 14: Number of chunks translated by GF and the percentage of the total chunks of the correponding type for the German grammars.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-Sts	84.21	76.66	83.52	15.07	3.44	14.26	23.23	12.44	44.35
GF-SaBs	**73.69**	**62.52**	**72.25**	**20.74**	**4.72**	**18.49**	**32.69**	**20.42**	**68.05**
GF-SaEs	75.58	64.75	74.49	19.69	4.50	17.75	31.23	18.75	63.78
GF-UnBs	74.21	63.04	72.77	20.39	4.67	18.24	32.26	20.12	66.97
GF-UnEs	76.00	65.12	74.91	19.48	4.47	17.61	30.94	18.54	63.06

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-Sts	13.41	12.91	8.42	12.91	2.82	69.03
GF-SaBs	**20.02**	**19.04**	**12.15**	**19.04**	3.82	**99.87**
GF-SaEs	19.02	18.52	12.13	18.52	**3.85**	97.89
GF-UnBs	19.77	18.81	12.04	18.81	3.79	98.75
GF-UnEs	18.83	18.40	12.10	18.40	3.83	97.30

Table 15: GF translation for the English-to-German language pair for the MAREC test set.

chunks can be translated, a half of the previous case. For the other systems results are comparable and around a 60% of the chunks have a GF translation. The percentages for every kind of chunk are also similar being RelPs the most translated elements and NPs the most numerous ones.

In this case also all the metrics in the automatic evaluation (Table 15 for MAREC and Table 16 for EPO_{MT}) favour a same system. The safe way of acquiring the lexicon is still the best, but starting with the core lexicon produces better translations than starting with the extended lexicon as happened for French. The difference is significant: more than 4 points of ULC and 1 of BLEU for example. So, irrespective of the online methodology to extract the lexicon (safe vs. unsafe) the core lexicon is the best resource to start with, followed by the extended lexicon and finally a unique static lexicon gives the poorest scores for the GF engine.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-Sts	79.20	70.18	78.81	25.67	4.68	13.42	25.72	11.39	48.53
GF-SaBs	**70.59**	**58.95**	**69.68**	**29.69**	**5.63**	**16.52**	**33.69**	**18.70**	**67.31**
GF-SaEs	73.20	61.45	72.42	28.56	5.39	15.66	31.36	16.86	62.26
GF-UnBs	70.75	59.10	69.83	29.64	5.62	16.48	33.62	18.59	67.06
GF-UnEs	73.32	61.57	72.54	28.53	5.38	15.64	31.33	16.77	62.10

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-Sts	12.92	12.97	8.72	12.97	2.87	70.66
GF-SaBs	**18.98**	**18.57**	**12.44**	**18.57**	**3.81**	**100.00**
GF-SaEs	17.61	17.70	12.32	17.70	3.80	96.43
GF-UnBs	18.88	18.49	12.42	18.49	3.80	99.62
GF-UnEs	17.53	17.67	12.31	17.67	3.79	96.22

Table 16: GF translation for the English-to-German language pair for the $\mathrm{EPO}_{\mathrm{MT}}$ test set.

4.4 Further improvements

When analysing the results (both here and with the hybrid outputs of Section 7) for the English-to-French translation one can conclude that the GF translator is difficult to improve. First, because the SMT system is very good already, and the GF cannot contribute with better translations and, second, because problems are usually not syntax related, so GF cannot bring novelties working on a word-to-word basis. On the other hand, there is more room for improvement in German assuming that one could obtain larger chunks in the first place. In this way the long distance dependencies - like complementation, predicates, could be resolved by the grammar and hence improve over the SMT translations which do not cope well with these phenomena. Some examples of these problems are given below.

Missing complements. Since sometimes there are elements between a verb and its object complement, the chunker may not recognise that the two should really belong together. This means two things – first, if the complement is a NP, it might not stand in the accusative case like it would if the VP consisted from the verb and the noun phrase together; and second, GF will not parse or translate verbs which have too many or too few arguments. By default, all verbs in the domain-specific dictionary used are two-arity verbs, i.e. they require a direct object. If this object is not found because it is separated from the verb, the whole phrase will not be translated. The same might apply to verb phrases with some other kind of complement. This is why it might make sense to introduce functions which parse verb phrases which would normally require to include a complement without it; in these cases it suffices to parse and translate the finite verb of the verb phrase, i.e. the string field of the verb.

A similar situation arises in cases where the verb is a copula with an adjectival or nominal complement. However, this is not so much a problem of the verb itself, since GF is able to translate the standalone copula in the sense of ‘to exist’, but of word order: VPs in German may have different constituent orderings, depending on the type of sentence. In subordinate clauses (which are essentially all of the verb phrases occurring in the patent claims, since those usually consist only of noun phrases), the constituents are in a different order, specifically, the finite verb in a verb phrase always comes last.

Long-distance agreement and agreement with embedded NPs. There are a few issues which could have not been solved so far with the chunk-based approach. One of the greatest strength of a rule-based system like GF is the handling of long-distance agreement, e.g. between subject noun phrase and predicate verb. However, there are cases where it cannot be reasonably determined which noun phrase the verb phrase or relative pronoun should agree with: In a claim like *the (...) dosage form of claim 1, which is a tablet (...)*, the relative pronoun could be agreeing with the subject noun *dosage form*, but theoretically also with the noun phrase embedded in the adverbial phrase: *claim 1.* In this example, the agreement is correct, but there are a number of dubious cases.

Another issue with even more room for ambiguity is the case where multiple noun phrases are case-governed by a preposition, e.g. *(...) a condition selected from gastric ulcers, duodenal ulcers, erosions of the stomach and esophagus, erosive gastritis (...)* where actually all NPs are governed by the preposition *from.* In the English and French grammar this obviously does not matter much, since these languages do not have as much morphologically marked agreement as German does. It is however not feasible to enforce preposition agreement over chunk boundaries, partly because oftentimes discerning is hard even for the human reader, and partly because noun phrases with adverbial attributes (*erosions of the stomach*) would greatly disturb the system.

5 General Lexical Resources

In addition to the domain specific lexicons that were tailored specifically to the patent corpus and have been explained in the previous section, we also wanted to create a general purpose lexicon that is reusable in other GF applications. This is also needed in the hybrid system where GF was used as a primary translation engine.

In the previous case, lexicons were build with the help of the resources obtained from in-domain parallel corpus, that is, lexical and phrase translation tables. Here, we need to build general lexicons so they are extracted from all-purpose resources such as WordNet and the Apertium dictionaries.

5.1 Uni-Sense Lexicons

We already had a comprehensive monolingual lexicon of about 43,000 words for English. This lexicon was originally based on the Oxford Advanced Learner’s dictionary [28] but it is now tuned to match with the syntactic categories used in the resource library. In MOLTO, we extended it further in three directions.

First of all, the lexicon was still small compared to lexical resources such as the Princeton WordNet [13] which provides a much bigger repository of English lexical data. We extended our lexicon with new lemmas extracted from WordNet. The result is a full-form English lexicon of around 65,000 lemmas. The slight difficulty here was that WordNet provides rich semantic information but no morphological information. This was solved by taking advantage of the fact that most irregular English words are already in the existing dictionary. For the new words we just derived the morphological table by using the morphological functions provided by the resource library. The accuracy of the paradigms was already evaluated in [8] and it has been shown that for English it is almost always possible to guess the fill-form table by using only the basic form. Although mistakes are possible it would take too much time to check all new words one by one.

Furthermore, we extended the lexicon with a list of prepositions which we collected from the Penn Treebank and from an article in Wikipedia about the English prepositions. The most important contribution of the latest is a list of multiword idiomatic prepositions which are harder to identify in the treebank.

Finally, we extended all verbs in the lexicon with information about their valency frames. We looked up every verb from the Penn Treebank and we reconstructed its valency frame from the context in the annotated sentences. After that the extracted information was added to the corresponding entry in the GF lexicon. The valency information is crucial when the lexicon is used in combination with the resource grammar for parsing and generation.

Having a good English lexicon is only the baseline but for translation we also need translation dictionaries, and we started building such resource for English-Bulgarian, English-Finnish, English-German, English-Hindi and English-Urdu by reusing existing open-source resources.

We extracted an English-German dictionary of 9,700 lemmas from Wiktionary. This dictionary provides full-form inflection tables for all German words found in Wiktionary and it shares one and the same abstract syntax with the English lexicon. This means that it can be used both as a morphological lexicon for German and as a translation dictionary from English to German. Unfortunately, as it could be seen, the coverage of this dictionary is much smaller, since it covers only 9,700 of the total 65,000 English entries.

The English-Bulgarian dictionary was bootstrapped from the translation dictionary for English-Macedonian, which we got from the Apertium system [14]. Since Bulgarian and Macedonian are closely related we were able to reuse this resource by automatically replacing some regular alternations between the two languages, followed by manual postediting, filtering and extensions. We also added some more lexical entries from the Universal WordNet [7]. Again, both resources do not provide morphological information, but we were able to complement them with the morphological lexicon from the Bulgarian spell checker in Open Office.

The parallel data for building the Finnish and Hindi dictionaries was taken from the corresponding WordNets [18, 23]. For Urdu the data was extracted from different resources including the Waseem Siddiqi's English-Urdu dictionary[14].

The following table summarizes the number of lemmas in the different dictionaries:

Bulgarian	Finnish	English	German	Hindi	Urdu
10,134	56,000	65,000	9,700	25,800	30,000

All this dictionaries share the same abstract syntax which makes it possible to use them as translation dictionaries. However, since we started from an abstract syntax originally designed for the monolingual English lexicon, those dictionaries have the severe problem that each English word must be unambiguously translated as exactly one word in the target language. For example the English-Hindi dictionary was built by simply selecting the first Hindi word which is essentially a random choice. As the words can have multiple senses, and it is often very hard to find one-to-one word mappings between languages, we also developed another abstract syntax which can accommodate several senses per lemma.

5.2 Multi-Sense Lexicons

A multi-sense lexicon is a more comprehensive lexicon and it contains multiple word senses with their translations to other languages. These multi-sense lexicons have been developed using data from the Princeton WordNet [13] and the Universal WordNet [7]. The original Princeton WordNet defines a set of word senses, and the Universal WordNet maps them to different languages. In this multilingual scenario, the WordNet senses can be seen as an abstract representation, while the lexical units can be seen as the concrete representation of those senses in different languages.

The Princeton WordNet is distributed in the form of different database files. For each of the four lexical categories (i.e. noun, verb, adjective, and adverb), two files named "index.pos" and "data.pos" are provided, where "pos" is noun, verb, adj and adv. Each of the "index.pos" files contain all words, including synonyms of the words, found in the corresponding part of speech category. While, each of the "data.pos" files contain information about unique senses belonging to the corresponding part of speech category, for our purposes, there were two possible choices to build an abstract representation of the lexicon:

1. To include all words of the four lexical categories, and also their synonyms (i.e. to build the lexicon from "index.pos" files)

2. To include only unique senses of the four categories with one word per sense, but not the synonyms (i.e. to build the lexicon from the "data.pos" files)

To better understand this difference, consider the words *brother* and *buddy*. The word *brother* has five senses with sense offsets "08111676", "08112052", "08112961", "08112265"

[14] Freely available for downloading and editing at
`http://www.scribd.com/doc/8509778/English-to-Urdu-Dictionary`

and “08111905” in the Princeton WordNet 1.7.1, while the word *buddy* has only one sense with the sense offset ’08112961’. Choosing option (1) means that we have to include the following entries in our abstract lexicon:

```
brother_08111676_N
brother_08112052_N
brother_08112961_N
brother_08112265_N
brother_08111905_N
buddy_08112961_N
```

We can see that the sense with the offset “08112961” is duplicated in the lexicon: once with the lemma *brother* and then with the lemma *buddy.* However, if we choose option (2), we end up with the following entries:

```
brother_08111676_N
brother_08112052_N
brother_08112265_N
brother_08111905_N
buddy_08112961_N
```

Since the file “data.noun” lists the unique senses rather than the words, there will be no duplication of the senses. However, the choice has an obvious effect on the lexicon coverage, and depending on whether we want to use it as a parsing or as a linearisation lexicon, the choice becomes critical. Currently, we choose option (2) for the following reason.

1. The Universal WordNet provides mappings for synsets (i.e. unique senses) but not for the individual synonyms of the synsets. If we choose option (1), as mentioned previously, we have to list all synonyms in our abstract representation. But, as translations are available only for synsets, we have to put the same translation against each of the synonym of the synset in our concrete representations. This will not gain anything but will increase the size of the lexicon and hence may have a negative impact on the processing speed.

Our abstract GF lexicon covers 91,516 synsets out of around 111,273 synsets in the WordNet 1.7.1. We excluded some of the synsets with multi-word lemmas. We consider them more of a syntactic category rather than a lexical category, and hence deal with them at the syntax level. Here, is a small segment of our abstract GF lexicon:

```
abstract LinkedDictAbs = Cat ** {

  fun consentaneous_00526696_A : A ;
  fun consecutive_01624944_A  : A ;
  fun consequently_00061939_Adv : Adv ;
  fun abruptly_00060956_Adv  : Adv ;
  fun consequence_09378924_N : N ;
  fun consolidation_00943406_N   : N ;
  fun consent_05596596_N  : N ;
  fun conservation_06171333_N : N ;
  fun conspire_00562077_V : V ;
  fun sing_01362553_V2  : V2 ;
  ........
  ........
 }
```

The first line in the above given code states that the module `LinkedDictAbs` is an abstract syntax. It extends another module labelled `Cat` which defines the morphological categories "A", "Adv", "N" and "V". These categories correspond to the "adjective", "adverb", "noun", and "verb" categories in the WordNet. However, note that in the GF resource library we have a much fine-grained subcategorization for verbs. We sub-categorize them according to their valencies i.e "V" is for intransitive, and "V2" for transitive verbs. The valency information we took from the monolingual English dictionary described in the previous section.
Each entry in the module is of the following general type:

```
fun lemma_senseOffset_T : T;
```

The keyword `fun` declares each entry as a function of the type "T". The function name is composed of lemma, sense offset and the type, where lemma and sense offset are the same as in the Princeton WordNet 1.7.1, while "T" is one of the lexical types in the GF resource library.

The abstract representation serves as a pivot for all concrete representations. We build the concrete representations for the different languages by using the translations obtained from the Universal WordNet data and the morphological paradigms in GF [8]. The Universal WordNet translations are tagged with a sense offset from WordNet 3.0[15] and also with a confidence score. As, an example consider the following segment form the Universal WordNet data, showing German translations for the noun synset with offset "13810818" and lemma *rest* (in the sense of *remainder*).

```
n13810818 Rest          1.052756
n13810818 Abbrand       0.954620
n13810818 Ruckstand     0.924376
n13810818 Restbetrag    0.662388
n13810818 Restauflage   0.446788
n13810818 Restglied     0.446788
n13810818 Restbestand   0.446788
n13810818 Residuum      0.409192
```

[15] However, in our concrete lexicons we match them to WordNet 1.7.1 for the reason that in our recent experiments, we are using these lexicons to build a free text translator. This translation system is using an external Word-Sense Disambiguator, which is based on WordNet 1.7.1. However, this can easily be mapped back and forth to other WordNet versions.

Here, each entry has the following general format.

```
posSenseOffset translation confidence-score
```

In cases, where we have more than one candidate translations for the same sense (as in the above case), we select the best one (i.e. with the maximum confidence score) and put it in the concrete grammar. Next, we give a small segment from the German concrete lexicon for the above given abstract lexicon segment.

```
concrete LinkedDictGer of LinkedDictAbs = CatGer ** open
 ParadigmsGer, IrregGer,Prelude in  {

  lin consentaneous_00526696_A =  mkA "einstimmig" ;
  lin consecutive_01624944_A =  mkA "aufeinanderfolgend" ;
  lin consequently_00061939_Adv =  mkAdv "infolgedessen" ;
  lin abruptly_00060956_Adv =  mkAdv "gech" ;
  lin consequence_09378924_N =  mkN "Auswirkung" ;
  lin consolidation_00943406_N =  mkN "Konsolidierung" ;
  lin consent_05596596_N =  mkN "Zustimmung" ;
  lin conservation_06171333_N =  mkN "Konservierung" ;
  lin conspire_00562077_V = mkV "anzetteln" ;
  lin sing_01362553_V2 = mkV2 (mkV "singen" ) ;
  ......
  ......
 }
```

The first line declares `LinkedDictGer` to be the concrete representation of the previously defined abstract representation (note the keyword `concrete` at the start of the line). Each entry in this representation is of the following general type:

```
lin lemma_senseOffset_T = paradigmName "translation" ;
```

The keyword `lin` declares each entry to be a linearisation of the corresponding function in the abstract representation. `paradigmName` is one of the morphological paradigms defined for German in the `ParadigmsGer` module. In the above code this are “mkA”, “mkAdv”, “mkN”, “mkV” and “mkV2” for the different lexical categories i.e. “adjective”, “adverb”, “noun”, “intransitive verb”, and “transitive verb”. “translation” in double quotes is the best possible translation obtained from the Universal WordNet. This translation is passed to a paradigm as a base word, which then builds a full-form inflection table. These tables are then used in the linearisation phase of the translation system.

Concrete lexicons for all other languages were developed using the same procedure. Table 17 gives some statistics about the coverage of these lexicons.

6 Robust Statistical Parsing

As an alternative to the integration between SMT and GF grammars, we also investigated the possibility to use the Resource Grammars Library for direct translation. Some preliminary results were known in advance, since Angelov, 2009 [1] gave the first hint that parsing

Language	Number of entries
Abstract	91,516
German	49,439
French	38,261
Finnish	27,673
Swedish	23,862
Hindi	16,654
Bulgarian	12,425

Table 17: Lexicon Coverage Statistics.

with the resource grammars is possible and actually quite efficient. Once the sentence is parsed, the constructed tree can be linearised back to another language which gives us a baseline translation service.

Unfortunately, the evaluation done in Ref. [1] was on a synthetic corpus with a very small lexicon. Furthermore, no attempt has been done to disambiguate between all possible readings of a sentence. Instead the parser just generated all readings in an arbitrary order. In the scope of MOLTO, this result was re-evaluated by doing a similar evaluation of the English grammar on a naturally occurring text from the Penn Treebank [25].

The new experiment is different in a number of ways. First of all the corpus is not synthetic any more which means that the different syntactic constructions are distributed with their usual probabilities. Second, this time the grammar was extended with the large English lexicon that we described in Section 5.1. Finally, the parser is augmented with a statistical model which is used in two ways. On one side it is used to rank the possible directions in the search space for the parser and thus it makes it possible to traverse only parts of the space. On the other side it helps with the disambiguation by picking the abstract syntax tree with the highest probability for a sentence. Parsing sentences with such a large and ambiguous grammar is a lot harder problem than the problem of parsing with the usual GF application grammars. Despite this the experiment confirmed that this is possible with a reasonable efficiency. Unfortunately we also realized that there is still a need for improvements before parsing unrestricted text becomes really flawless.

The first problem is that we had to restrict the length of the input sentences to 25 tokens since otherwise the parser just fails due to the excessive memory consumption. This limit is still quite low and it is not unusual to have longer sentences in a naturally occurring text. Unfortunately this is the current limit of the latest statistical parser with the English grammar. For comparison, we measured that the statistical parser is significantly faster and with lower memory requirements than the previous non-statistical version. The average parsing time for both parsers is shown on Figure 5. It is obvious that the new parser is significantly faster. It also has lower memory requirements. We had to put a limit of 10 tokens for the non-statistical algorithm since after that the memory requirement becomes too high. On the same corpus the statistical version can cope with lengths up to 25. The

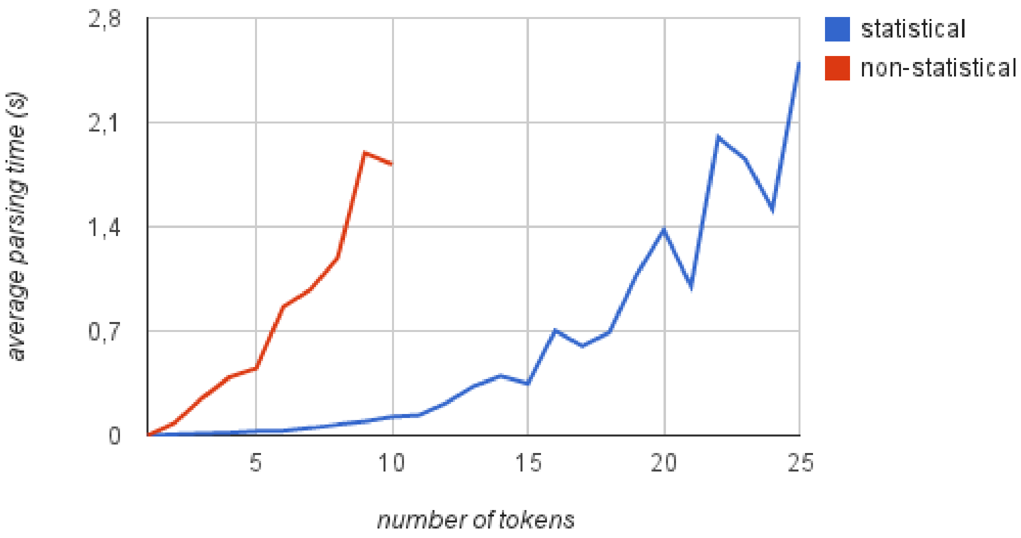


Figure 5: Average parsing time for the statistical and for the non-statistical GF parser.

evaluation was done on an Intel Core i5 processor with 8Gb RAM.

We also wanted to compare our statistical parser with other existing parsers based on a similar formalism. Although GF is a high-level programming language, during the compilation, the source language is reduced to a well-known and simple formalism: Parallel Multiple Context-Free Grammar (PMCFG, [38]). Rparse [19] is a state-of-the-art training and parsing system for PMCFG. It is written in Java and developed at the Universities of Tübingen and Düsseldorf, Germany. Rparse can be used for learning probabilistic PMCFGs from discontinuous treebanks and for parsing new sentences with the learned grammars. In our evaluation we used Rparse to extract PMCFG grammars from the discontinuous German Tiger Treebank [6]. After that we used both Rparse and the GF parser for parsing new sentences with the extracted grammars. In all cases both parsers returned identical results, but as it can be seen on Figure 6, our parser significantly outperforms Rparse. For instance, for sentences of length about 25 tokens our parser becomes about 100 times faster than Rparse.

Our algorithm has also other interesting properties. Currently GF is the only system which supports the full power of PMCFG. For instance, Rparse requires that the grammar is binarised and linear, which means that it only supports linear context-free rewriting systems (LCFRS) in binary form. This restriction is not necessary in our case. The grammar learner in Rparse actually first learns a non-binary grammar which after that has to be binarised, before it can be used for parsing in their system. Since this is not a problem for the GF parser, we evaluated Rparse only on the binarised grammar, and the GF parser on both the binarized and the non-binarised grammar. Figure 6 shows that GF actually performs even better on the non-binary grammar than on the binary grammar.

The details about the parsing algorithm and the evaluation are published in Ref [3]. The experiment told us that we have a statistical parser which is beyond the state of the art for

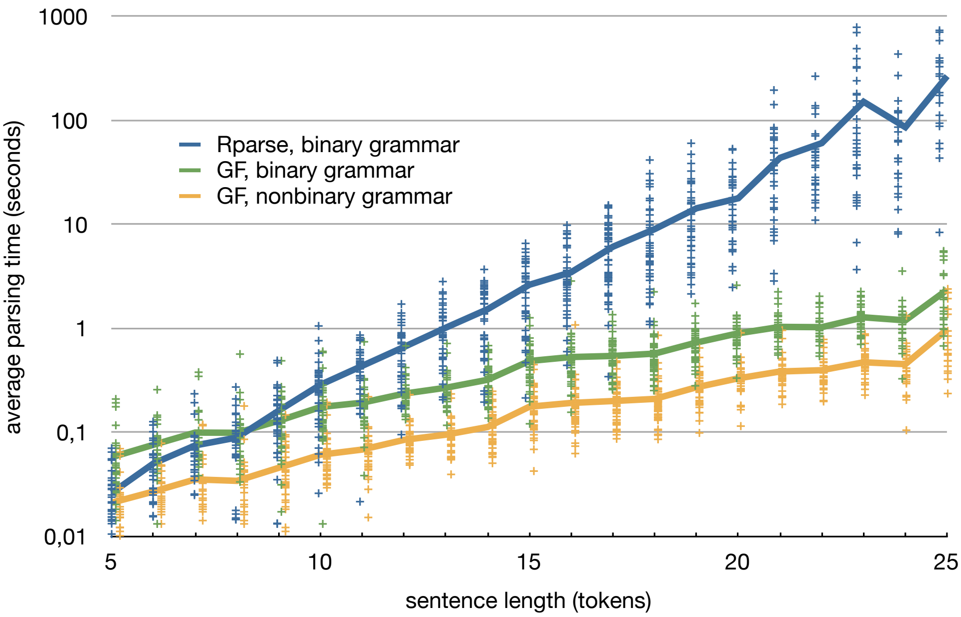


Figure 6: Parsing time for Rparse and the GF parser.

PMCFG. Its efficiency is actually quite satisfactory on the automatically learned German grammar. The fact that it is still not good enough with the English Resource Grammar is partly surprising. We still have to analyse this further, but one possible explanation might be that the English grammar is a lot more complex than the derived German grammar. For instance the English grammar uses a lot more discontinuities and empty linearisation rules in order to achieve language independent abstract syntax.

The second problem that we faced in the effort to make robust GF translator was to cope with the limited coverage of the English grammar. The basic idea behind the strategy for robustness is a generalization of Stolcke's idea [40] for introducing wildcard states. While his approach works for context-free grammars, we extended it to PMCFG. The outcome is that when the sentence is not fully parseable then the parser just returns the list of parseable chunks which score best according to the statistical model. From translation perspective this means that when the sentence is not in the scope of the grammar, then we can translate only some parts of it.

So far, we have not said much about the training data for our statistical model. The data is basically the Penn Treebank, but in order to make it useful, we had to transform it into a set of abstract syntax trees compatible with the English Resource Grammar. We developed a set of transformation patterns which map fragments of the annotations in the original treebank into fragments of abstract syntax trees. Whenever the annotation fragment cannot be represented with the grammar, the transformation simply generates a placeholder (?) and continues with the rest of the sentence.

An example sentence from the Penn Treebank and its corresponding GF tree are shown on Figure 7. Here we can see that a lot has been recovered but there are also three cases where the mapping has failed. First of all, it is not possible to construct the phrase:

```
Pierre Vinken, 61 years old, will join the board as a nonexecutive
director, Nov.  29
```

with the resource grammar for two reasons: the grammar have no constructions for building an adjectival phrase like "61 years old" and there is no way to attach an adjectival phrases to a proper name. Furthermore, the grammar does not support time adverbials like "Nov. 29" unless they are introduced with a preposition. Finally, the adjective "nonexecutive" is missing in the lexicon, so we get yet another placeholder.

We looked at the generated trees and whenever a grammar function has been recovered we always found that it is the correct one. However, there were situations where none of the patterns matched and then we got a placeholder, although the manual examination showed that the sentence is actually expressible in the grammar. Some of the failures were also due to errors and inconsistencies in the original annotations in the treebank but there are also cases where the patterns are just not comprehensive enough. Since writing exhaustive transformation patterns would be very difficult, we just manually completed some of those trees. Currently there are 7,144 sentences for which we are sure that they are as complete as possible with respect to the current coverage of the grammar. This number also includes 3,243 completely converted sentences.

Although the conversion is not complete, the automatic mapping gave us a way to estimate the coverage of the grammar on the corpus. However, the statistics should be taken only as estimations because they only reflect the current state of the corpus. We expect that the numbers for the complete conversion would be higher.

There are many ways to compute the coverage of the grammar. The simplest one is to calculate the percentage of nodes in the abstract trees that are filled in with function symbols instead of placeholders. A plain counting shows that in average 94.80% of the nodes are filled in with functions, and the remaining 5.20% are placeholders. However, this percentages include both placeholders which are due to missing syntactic constructions and those that are due to unknown words. It is a good idea to separate them since in any kind of processing, unknown words are easier to handle than unknown syntax. If we pretend that the placeholders on the leaves of the trees are actually known words we get the much higher coverage of 96.77%.

Another way to evaluate the coverage is to look at how many sentences were fully converted and how many have only one, two, three or more placeholders. As a whole, out of 49,208 sentences, 3,243 sentences (6.58%) were fully converted to abstract trees. Again, if we exclude the unknown words, we get much higher numbers - 5,830 sentences or 11.85%. The overall distribution of the number of sentences with a given number of placeholders is shown on Figure 8. It can be seen that the majority of sentences is clustered in the region from one to four placeholders per sentence. More concretely the expected average number is 3.34 with a standard deviation of 2.25. If we ignore the unknown words we get a sharper curve which gives an expected value of 2.07 placeholders with a deviation of 1.46.

```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
```

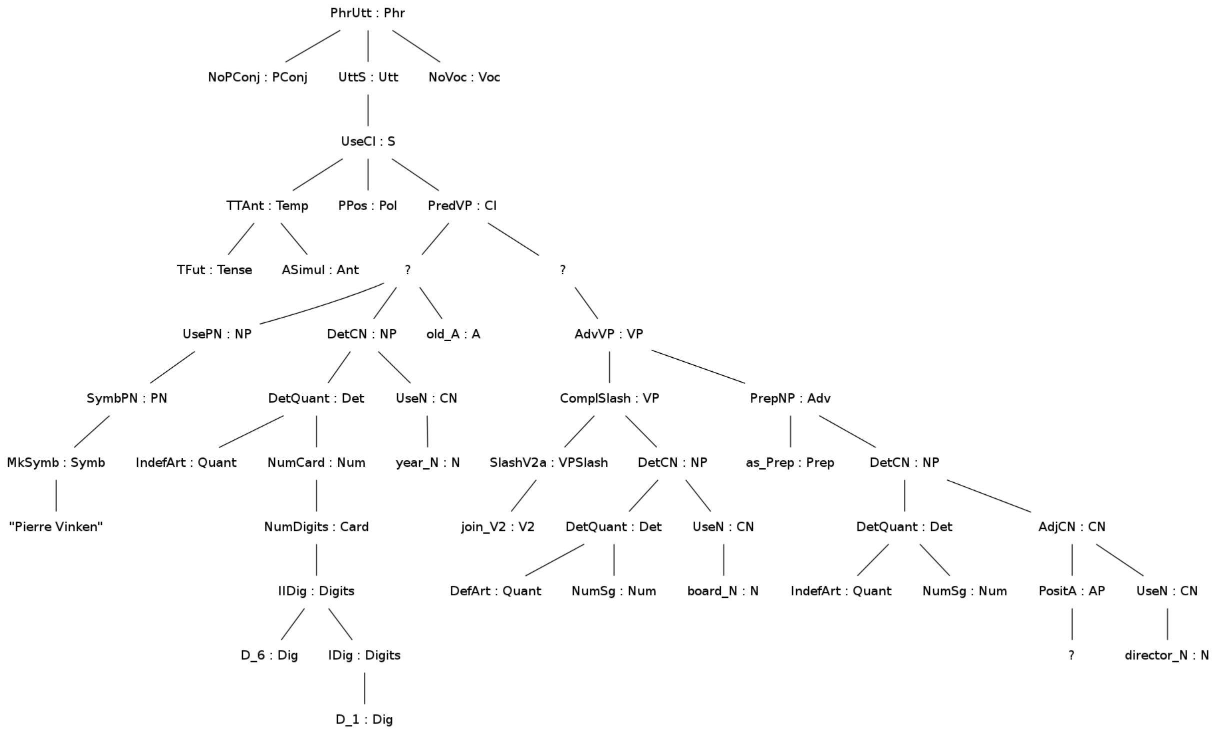


Figure 7: An example Penn Treebank tree (above) and the corresponding abstract tree in the English Resource Grammar (below)

Although, the plain percentage of recovered function names is quite high, the percentage of sentences that are actually parseable is quite low. This shows that grammar can explain most of the syntactic constructions in the sentence, but still given the usual distribution of sentence lengths in the Penn Treebank, there is still a high chance to encounter an unknown construction in most of the sentences. In general the chance to find new constructions is higher in a longer sentence. This points to an alternative measure for the coverage. We

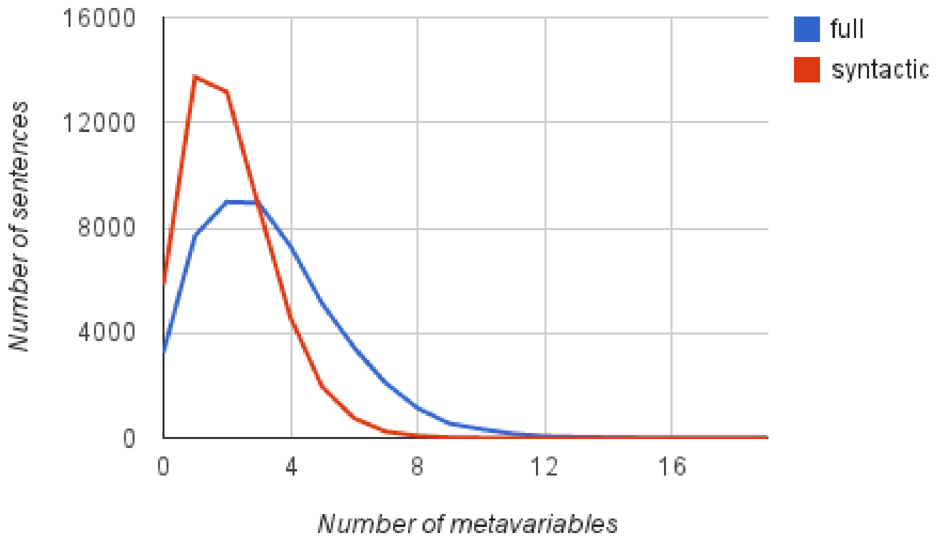


Figure 8: The distribution of the number of sentences with a given number of placeholders.

can compute the ratio between the number of tokens in a sentence and the number of placeholders. We found that in average it can be expected to encounter a new placeholder once in every 8.78 tokens, or if we ignore the unknown words in every 12.86 tokens. This correlates well with the observation that the average length of the fully converted sentences is 11.43 tokens although there are sentences of length up to 67 tokens. In general the grammar works better on shorter sentences. The same was also confirmed in the SMT+GF experiments where the coverage was improved by parsing chunks instead of full sentences (Section 4).

The new statistical parser in GF in combination with the Resource Grammars Library and the large multilingual lexicons, opened the possibility to use GF as a standalone translation system which may or may not be complemented with SMT. However, we still think that using GF as the primary translation engine still has not reached its full potential. As we will see in the following section our translation scores are still pretty low.

6.1 Robust parsing performance

With the help of the new statistical parser in GF, we built a baseline translation system where the input sentence is parsed from English to an abstract syntax tree and after that the tree is directly linearised to a target language. To evaluate different properties of the system we have conducted two different rounds of the experiments. Details are given in the following subsections.

Languages	**System**	**BLEU**	**WER**	**PER**	**TER**
English-Hindi	Google	30.96	55.85	42.96	52.98
	GF	34.30	60.62	50.84	59.19
English-Finish	Google	29.11	51.07	46.79	48.40
	GF	34.93	45.99	39.04	45.99
English-German	Google	62.70	24.93	19.10	22.81
	GF	50.13	31.03	29.71	31.03

Table 18: Evaluation results with pre-parsed sentences.

6.1.1 Round 1

In the first round, we took a set of pre-parsed sentences (i.e. parse trees) and linearised them to Hindi, German, and Finnish by using the resource grammar for the corresponding language. Since we did not use the GF parser (neither the regular GF parser, nor the robust one) in this round, the purpose it to show that once we have a correctly parsed and disambiguated trees, our base-line translation system, with some pre-supposed limitations, can be competitive with state-of-the-art engines such as Google Translate. We evaluated our translations against the gold-standard translations produced by correcting the automatic translations obtained from either Google Translate or GF. In this way we prepared a test corpus which contains 100 sentences, manually corrected by native speakers. To avoid any biasing, 50 of those sentences were initially produced by Google Translate and 50 by GF. Table 18 shows the BLEU, WER, PER, and TER evaluation scores for the different language pairs on the test corpus. WER, PER, and TER all are error rates, so small values are better, while BLEU is a performance score, so higher value is better. From the table we can see that our baseline translation system got consistently better BLEU scores than Google. However, Google has slightly better scores for the error rates on the English-Hindi language pair. Still, the comparison with Google is not completely fair, because we are evaluating the systems on in-grammar sentences.

6.1.2 Round 2

In this round, we took a set of random sentences from the Penn Treebank data, we parsed them by using the statistical parser and then linearized them to German, and Hindi. Table 19 shows the evaluation results. We can see that our translation system performed better than Google for German, but for Hindi the Google scores are better. One reason could be already known issues with the Hindi resource grammar. For this round we only considered fully parsed sentences and left the partially-parsed sentences out of the evaluation test. This is why, the comparison with Google is still not completely fair.

Languages	System	BLEU	WER	PER	TER
English-Hindi	Google	27.01	60.31	44.07	56.96
	GF	21.72	70.52	61.05	65.98
English-German	Google	41.55	41.76	38.46	38.74
	GF	45.64	36.82	32.14	36.81

Table 19: Evaluation results with statistical parsing.

7 Hybrid Systems

As it has been seen, the grammar-based translator already makes use of the SMT system trained on patents to translate the GF English lexicon. Although it already uses of hybridisation techniques, we consider this first approximation as a baseline for the more advanced hybrid systems. The reason is that even the vocabulary is disambiguated towards the biomedical domain thanks to the hybridisation, still there are non-parseable chunks with unknown vocabulary in the lexicon that cannot be translated using the grammar. That is to say, the system is not able to translate robustly a whole test set. The percentage of sentences that can be completely translated from beginning to end by GF is 6.9%.

In the description of work three baseline systems were proposed: the individual SMT system, the individual GF system, and a naïve cascade combination where a sentence is translated by GF whenever is possible and by SMT otherwise. After the development of the systems we have defined two different baseline systems: the SMT system and the first hybrid GF translator. The latter is already a hybrid baseline and the naïve combination of both does not provide new information since more than 93% of the sentences would be translated by SMT.

To gain robustness in the final system the output of the GF translator is used as *a priori* information for a higher level SMT system. An SMT system trained in the same way as the SMT baseline is fed with these GF phrases and it is the way the phrases interact with the SMT ones what defines two families of hybrid models:

Hard Integration (HI). Phrases with GF translation are forced to be translated this way. The top decoder can reorder the chunks and translates the untranslated chunks, but there is no interaction between GF and pure SMT phrases. From all the possible SMT phrases that can fill the holes left by GF, the language model chooses those that make the output more fluent, understood as the model of fluency given by the training corpus.

Soft Integration (SI). Phrases with GF translation are included in the translation table with a certain probability so that the phrases coming from the two systems interact. Probabilities in the SMT system are estimated from frequency counts in the usual way; the probabilities in the GF system are assigned to divide phrases in two or three groups according to the confidence on the translation as explained in Section 4.3. The GF score is divided homogeneously among the four lexical features of the translation table. So, a score of 4 means a score of 1 for each feature (generative and discriminative lexical translation probabilities $lex(f|e)$ and $lex(e|f)$, and generative and discriminative translation models $P(f|e)$ and $P(e|f)$). This value does not imply a sure translation because other features such as language model, penalties and reordering also interact, but it is given a high value so that SMT phrases at most can tie. In the same way, a score of 1 is converted into a 0.25 for each feature, and the most unsafe translations, those with score 0.2, have a minimum contribution of 0.05 per feature. This small value makes them very difficult to be chosen but still complement the translation in case there is not better choice coming from the SMT.

Under this perspective, the GF translator can be seen as a Soft Integration model led by a GF translator instead of being led by the SMT. The patents grammar parses the source sentence and complements its translations by the ones given by the SMT translation of the lexicon. These three integrations are evaluated on the patents test set both automatic and manually.

7.1 Integration led by SMT

We analyse here the hybridisation between the GF translators from Section 4 and an SMT system. With the five different grammars, the two kinds of integration, and the degree of interaction one can define 40 hybrid systems. By the degree of interaction we refer to the relevance of GF in the top SMT translator: GF can be used only at decoding time or also for tuning the parameters. So, in this case, the weights of all the SMT components (Eq. 1) are adjusted following the same methodology applied in test time. The development for the hard (soft) integration system is done with a hard (soft) integration of the GF translations of the development set with the SMT translation table. The latter case is indicated by *dev* in the system's names, hard and soft integration are marked as *HI* and *SI* respectively, and the nomenclature for the GF grammar is that of Table 10.

Table 29 in Appendix A summarises the nature of these 40 systems. The names defined in this table are used throughout the document. In order to ease the reading, the complete automatic evaluation of these engines is included as Appendix A. This section shows the analysis for the most representative systems and for those with the best performance.

In general, the best results are obtained with the static lexicon, a soft integration and with GF used in development. The other options are less relevant to the final results, although is some cases differences are not significant.

As expected, the largest difference between the soft and hard integration is the number of GF phrases in the final translation. For French, the MAREC test set parsed with `Genia` is divided in 10,306 grammatical chunks[16] (or 10,456 for the unsafe version). As it has been seen, only 4,255 chunks can be parsed by `Genia` with the static lexicon, a 41% of the total.

[16] SMT phrases can be shorter or longer than grammatical chunks.

	GF	**SMT**	**BOTH**	Total
HIdev-Sts	1,486 (34.92%)	0 (0.00%)	2,769 (65.08%)	4,255
HIdev-SaBs	3,228 (52.65%)	1 (0.02%)	2,902 (47.33%)	6,131
HIdev-SaEs	3,242 (50.57%)	1 (0.02%)	3,168 (49.42%)	6,411
HIdev-Stm	1,435 (33.73%)	0 (0.00%)	2,820 (66.27%)	4,255
HIdev-SaEm	2,683 (41.85%)	1 (0.02%)	3,727 (58.13%)	6,411
SIdev-Sts	250 (5.88%)	1,897 (44.58%)	2,108 (49.54%)	4,255
SIdev-SaBs	323 (5.27%)	3,656 (59.63%)	2,152 (35.10%)	6,131
SIdev-SaEs	354 (5.52%)	3,737 (58.29%)	2,320 (36.19%)	6,411
SIdev-Stm	230 (5.41%)	1,936 (45.50%)	2,089 (49.10%)	4,255
SIdev-SaEm	438 (6.83%)	3,269 (50.99%)	2,704 (42.18%)	6,411

Table 20: Origin system of the chunks translated by several hybrid systems which have both a GF and an SMT translation option. The remaining chunks and the punctuation tokens can only be translated by the SMT system.

	GF	**SMT**	**BOTH**	Total
		SIdev-Stm		
VP	66 (13.50%)	222 (45.40%)	201 (41.10%)	489
NP	85 (4.85%)	776 (44.32%)	890 (50.83%)	1,751
AdjP	0 (0.00%)	27 (55.10%)	22 (44.90%)	49
AdvP	79 (5.41%)	791 (54.18%)	590 (40.41%)	1,460
RelP	0 (0.00%)	120 (23.72%)	386 (76.28%)	506
Total	*230 (5.41%)*	*1,936 (45.50%)*	*2,089 (49.10%)*	*4,255*
		SIdev-SaEm		
VP	94 (10.77%)	495 (56.70%)	284 (32.53%)	873
NP	204 (7.26%)	1,332 (47.39%)	1,275 (45.36%)	2,811
AdjP	2 (2.30%)	31 (35.63%)	54 (62.07%)	87
AdvP	138 (6.77%)	1,248 (61.27%)	651 (31.96%)	2,037
RelP	0 (0.00%)	163 (27.03%)	440 (72.97%)	603
Total	*438 (6.83%)*	*3,269 (50.99%)*	*2,704 (42.18%)*	*6,411*

Table 21: As in Table 20 but results are broken down according to the type of chunk. The above figures correspond to the SIdev-Stm hybrid system; below, correspond to SIdev-SaEm.

This number grows up to around 60% for dynamic lexicons. The hard integration system uses almost all of them in the final translation. Still, the chunks that are not covered by GF are translated with SMT phrases and, besides, some of the GF translations can also be built from SMT phrases. So, at the end, only around 15% of the grammatical chunks are translated thanks to the GF grammar with static lexicons and a 30% with the dynamic ones. An additional 30% of chunks which already have a SMT translation are collapsed towards the GF translation.

Numbers are much lower for the soft integration. Table 20 shows the comparison for a representative subset of systems. In this case between $5-7\%$ of the chunks are uniquely translated by GF. The language model which, together with the word penalty, has the highest weight in the final score of a translation, is favouring translations with structures similar to those seen in the training data. So, when the final decoding has freedom to choose, it reproduces the most frequent structures. In some cases that simply favours a correct SMT translation in front of another correct GF translation; in some others, a correct translation is lost because the language model is not seeing at large distances.

According to the type of chunk, verb phrases (VP) are the ones with a higher percentage of GF translations in the final output. However, VPs only represent between $10-15\%$ of the total. One can read the distribution per type of chunk in Table 21 for two of the hybrids: SIdev-Stm and SIdev-SaEm. Noun phrases (NP) and adverbial and prepositional phrases (AdvP) are more abundant in the test set: structures such as "`according to`" or "`of said combination/compound...`" are widely repeated through patents. This high frequency also seen in the SMT training set makes that half of the times the translation chosen is that from the SMT system and, in more than 95% of the cases the final translation is already within the original SMT translation table (SIdev-Stm). GF can be helpful in translating chunks for which agreement is crucial (VP and specially RelP and AdjP). As also seen in Table 21, RelPs and AdjPs are not translated by GF. However, in 45% and 76% of the chunks respectively the translation is both given by GF and SMT, and GF is collapsing the solution towards the one that does a correct agreement. That is, the translation of the pure SMT system is modified because of the reinforcement of another option by GF. The most evident example is agreement in gender and number. Contrary to English, French adjectives and nouns agree in gender and number and relative pronouns agree with their relative. This is taken into account by construction in GF so that mistaken SMT translations such as "le médicament séparée" is correctly translated as "le médicament séparé" (`the separate medicament`) or "composition pharmaceutique selon la revendication 1, dans lequel" is correctly translated as "composition pharmaceutique selon la revendication 1, dans lequelle" (`the pharmaceutical composition of claim 1, wherein`).

The small percentage of chunks that GF is modifying with respect to SMT translations makes a large improvement in the automatic evaluation of the translation difficult. Table 22 shows the selection of lexical and syntactic metrics applied on MAREC test set and Table 23 for the $\mathrm{EPO}_{\mathrm{MT}}$ test set. The first horizontal block corresponds to the baseline systems, the second one displays the results for the hybrid systems with the best performance, and the last block shows two external translators to compare our system with the state-of-the-art.

For all the metrics the SMT system beats the GF one in a significant way. This is mainly due to the coverage, SMT is able to translate the whole sentence which is not the case of GF. However, GF is able to deal with some grammatical issues that cannot be recovered statistically as seen before.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-SaEs	66.62	51.14	64.47	21.69	5.17	18.87	35.36	26.20	26.17
SMT	27.16	18.00	25.57	62.40	9.95	44.99	75.71	72.94	84.97
SIdev-Sts	26.90	17.71	25.33	62.82	10.01	45.31	76.11	**73.63**	85.55
SIdev-Stm	**26.81**	**17.69**	**25.23**	**63.05**	**10.03**	**45.57**	**76.21**	73.49	**85.73**
Bing	39.78	28.14	37.70	45.73	8.64	32.67	61.55	58.50	64.68
Google	32.46	20.82	30.54	57.86	9.70	40.28	71.56	71.40	78.87

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-SaEs	25.92	26.83	21.06	25.21	1.97	46.00
SMT	63.71	65.91	**50.92**	47.54	**3.60**	**99.70**
SIdev-Sts	**64.05**	**66.36**	50.25	**47.62**	3.59	**99.70**
SIdev-Stm	63.61	66.21	49.97	47.55	3.59	99.35
Bing	49.65	51.93	36.95	40.34	3.27	80.79
Google	60.17	62.56	46.75	45.48	3.46	94.33

Table 22: Comparative table for the English-to-French language pair for the MAREC test set.

Notice that the table of best systems does not include any of the hard integration models (see Appendix A for the results). As explained, the hard integration of the translations does not allow them to interact. GF translations are always used and the statistical decoder reorders them and completes the translation with its own phrase table. Results in this case are below those of the SMT system because the system is being forced to use the high quality translations together with translations of elements not considered. Just to give an example, GF would highly benefit from incorporating a grammar to deal with compounds and numbers. Currently these elements typical of the domain are not specifically approached. Including multiple GF translations enhances the performance but, still, the pure SMT is better. A similar thing happens when considering the extended base lexicon for the runtime versions instead of the base one: the effects are visible for hard integration but are lost for soft integrations.

Another trend one can observe in the Appendix is the fact that using GF for tuning the weights of the final SMT decoder improves the performance. Improvements are not huge, but consistent through all metrics, test sets and language pairs.

The best translators are hybrids with a soft integration of the systems, the one giving more freedom to the combination. The combination of all the phrases improves the translations according to all the lexical metrics considered for the MAREC test set. Only the syntactic metrics CP-STM-9 and SP-pNIST-5 prefer the statistical system in front of the hybrids. The best system, SIdev-Stm, is above the SMT in 0.65 points of BLEU and 0.50 of METEOR for example. The mean of metrics, ULC, separates the hybrid from the

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-SaEs	61.92	50.35	60.48	25.36	5.81	17.42	36.26	24.97	28.23
SMT	30.08	**19.21**	**28.35**	61.08	9.94	40.75	72.27	69.90	82.59
SIdev-Sts	**29.95**	19.26	28.51	61.28	9.99	**40.94**	**72.43**	**70.66**	**82.91**
SIdev-Stm	30.14	19.26	28.67	61.15	9.96	40.91	72.25	70.48	82.69
Bing	42.44	30.74	40.01	44.21	8.59	28.02	57.75	57.44	61.16
Google	30.26	20.76	28.50	**61.43**	**10.28**	38.44	71.15	70.61	81.85

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-SaEs	25.94	28.04	22.46	36.29	4.60	46.34
SMT	62.90	66.55	**52.95**	73.64	7.99	99.58
SIdev-Sts	62.91	**67.21**	52.10	**73.90**	8.02	**99.61**
SIdev-Stm	**62.96**	67.04	52.23	**73.87**	8.00	99.56
Bing	48.54	52.11	37.43	60.96	7.02	79.01
Google	60.51	65.11	49.74	73.37	**8.04**	97.24

Table 23: Comparative table for the English-to-French language pair for the EPO_{MT} test set.

SMT in 0.76 points for the lexical metrics but shows comparable values for the syntactic metrics. The state-of-the-art systems, Google and Bing, are below these devoted engines.

For the EPO_{MT} test sets results are not so consistent. A hybrid system, SIdev-Sts, is still the best according to the combination of metrics, but the variation is wider. BLEU and NIST (lexical and syntactic) prefer Google's translations, although at the end ULC ranks the system after all the hybrids and even after the SMT. In this case, the SMT and the hybrids are closer, and the evaluation is better for the SMT with two metrics, PER and TER.

As a final remark it is seen that the best hybrids are built with a static lexicon. A static lexicon has less but more probable translations. From the three levels of confidence there are no chunks with a low score. Besides, the number of chunks with an intermediate score is severely reduced by more than a half, almost all the cut in the number of chunks comes from here because the system keeps the most sure translations. The distinction between the safe and unsafe runtime versions only affect about a 6% of the translated chunks, so differences on the final translations due to this change are minimum.

The behaviour for the English-to-German translator is similar, but with one main difference, the size of the static lexicon. As a result, the family of models with the static lexicon only have 2,264 chunks translated by GF for the MAREC test set and 2,242 for the EPO_{MT} one, this is half of the number that one has for French and the effect on the final performance is visible.

At the end, around a 20% of the chunks are uniquely translated by GF in the final

	GF	**SMT**	**BOTH**	Total
HIdev-Sts	518 (22.88%)	0 (0.00%)	1,746 (77.12%)	2,264
HIdev-SaBs	2,253 (37.51%)	1 (0.02%)	3,753 (62.48%)	6,007
HIdev-SaEs	2,742 (44.88%)	1 (0.02%)	3,366 (55.10%)	6,109
HIdev-Stm	483 (21.33%)	0 (0.00%)	1,781 (78.67%)	2,264
HIdev-SaEm	2,355 (38.55%)	1 (0.02%)	3,753 (61.43%)	6,109
SIdev-Sts	28 (1.24%)	1,171 (51.72%)	1,065 (47.04%)	2,264
SIdev-SaBs	253 (4.21%)	3,266 (54.37%)	2,488 (41.41%)	6,007
SIdev-SaEs	169 (2.77%)	3,686 (60.34%)	2,254 (36.90%)	6,109
SIdev-Stm	37 (1.63%)	1,015 (44.83%)	1,212 (53.53%)	2,264
SIdev-SaEm	229 (3.75%)	3,421 (56.00%)	2,459 (40.25%)	6,109

Table 24: As Table 20 for the English-to-German language pair.

	GF	**SMT**	**BOTH**	Total
		SIdev-Stm		
VP	3 (1.61%)	133 (71.50%)	50 (26.88%)	186
NP	11 (1.10%)	500 (49.95%)	490 (48.95%)	1,001
AdjP	1 (4.76%)	4 (19.05%)	16 (76.19%)	21
AdvP	22 (4.00%)	313 (56.91%)	215 (39.09%)	550
RelP	0 (0.00%)	65 (12.85%)	441 (87.15%)	506
Total	*37 (1.63%)*	*1,015 (44.83%)*	*1,212 (53.53%)*	*2,264*
		SIdev-SaEm		
VP	16 (2.16%)	653 (88.01%)	73 (9.84%)	742
NP	102 (3.65%)	1,291 (46.17%)	1,403 (50.18%)	2,796
AdjP	4 (4.12%)	58 (59.79%)	35 (36.08%)	97
AdvP	107 (5.72%)	1,342 (71.73%)	422 (22.55%)	1,871
RelP	0 (0.00%)	77 (12.77%)	526 (87.23%)	603
Total	*229 (3.75%)*	*3,421 (56.00%)*	*2,459 (40.25%)*	*6,109*

Table 25: As Table 21 for the English-to-German language pair.

translation for the hard integration and less than a 2% with the soft integration. These percentages are shown in Table 24 for the MAREC test set. Another noticeable point is that for the runtime versions there is also a decay in the number of chunk translations that are chosen from GF. For a hard integration this number is always below 45% (to be compared with 53% for French) and for a soft integration below 4.5% (7% for French). The best grammar, GF-SaBs, is the one with the larger contribution into the hybrid with a 4.2%, which, still is lower than the smallest GF contribution for French. However, in this

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-SaBs	73.69	62.52	72.25	20.74	4.72	18.49	32.69	20.42	26.09
SMT	30.93	**22.82**	29.33	57.59	9.40	42.98	**67.84**	63.14	84.81
SIdev-Sts	31.06	22.95	29.45	**57.70**	9.44	43.14	67.73	63.22	84.86
SIdev-Stm	31.59	22.97	29.97	57.35	9.41	42.82	67.43	62.91	84.35
SIdev-SaEs	**30.80**	23.39	**29.24**	57.40	**9.48**	**43.38**	67.31	**63.47**	**84.88**
Bing	53.94	40.49	51.71	36.30	6.95	27.95	51.27	42.74	54.26
Google	40.20	26.99	38.29	53.43	9.01	39.89	63.34	62.35	77.56

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-SaBs	20.02	19.04	12.15	19.04	3.82	40.63
SMT	52.89	**51.62**	32.99	**51.62**	**6.95**	**99.84**
SIdev-Sts	52.98	50.83	32.62	50.83	6.89	98.85
SIdev-Stm	52.81	50.71	32.39	50.71	6.89	98.54
SIdev-SaEs	**53.04**	51.13	**33.16**	51.13	**6.95**	99.60
Bing	35.77	34.87	20.50	27.30	5.47	65.66
Google	49.88	47.23	29.71	47.23	6.23	91.24

Table 26: Comparative table for the English-to-German language pair for the MAREC test set.

case, there is an increment on the number of chunks whose translation appears in both systems.

Regarding the type of chunk, NPs and AdvPs are the more numerous in the output and also the ones with a higher percentage of GF translations, but AdvP and VP are mostly translated by SMT phrases. RelPs are never translated by GF alone, because the number of translation options is small and they always appear in the translation table. For example "`which`" can be translated as "der", "die" or "das", GF always does the correct agreement when gets a translation and so, it helps to choose between the three options from the translation table.

The low presence of GF chunks in the final translation explains the results of the automatic evaluation. Tables 26 and 27 show the scores for the set of lexical and syntactic metrics for the MAREC and EPO$_{\mathrm{MT}}$ test sets respectively. The SMT and the best hybrid are close in scores. SIdev-StBs and also SIdev-SaEs are better according to most lexical metrics, whereas the SMT system is preferred by 4 out of the 6 the syntactic metrics. This is true for the *in-domain* set; for the EPO$_{\mathrm{MT}}$ one, the best hybrid system is always better than the SMT. However, in this case, Google obtains a higher performance according to NIST, METEOR and ROUGE.

Some general comments already mentioned in the English-to-French translation apply also for German. That is, a soft integration where GF has been used both in development

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-SaBs	70.59	58.95	69.68	29.69	5.63	16.52	33.69	18.70	29.12
SMT	35.32	25.59	33.47	58.46	9.50	37.86	63.44	61.12	81.04
SIdev-Sts	**34.76**	25.56	**32.92**	**58.93**	9.59	**38.33**	63.74	61.31	**81.72**
SIdev-Stm	35.17	**25.28**	33.22	58.55	9.56	37.88	63.59	61.56	81.40
SIdev-SaEs	34.86	26.22	32.94	58.13	9.60	38.31	62.98	61.44	81.27
Bing	53.24	38.46	50.78	42.34	7.62	24.96	50.03	42.40	55.74
Google	38.06	25.71	35.87	57.63	**9.77**	37.35	**63.95**	**64.01**	80.78

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-SaBs	18.98	18.57	12.44	18.57	3.81	38.76
SMT	52.07	50.15	32.75	68.17	6.64	98.89
SIdev-Sts	52.32	**50.46**	**33.16**	**68.76**	**6.75**	**99.86**
SIdev-Stm	52.04	49.92	32.15	68.21	6.69	98.58
SIdev-SaEs	52.13	50.06	32.69	68.23	**6.75**	99.19
Bing	35.85	34.65	21.06	60.16	5.45	73.68
Google	**52.68**	49.39	31.75	67.12	6.38	97.14

Table 27: Comparative table for the English-to-German language pair for the EPO_{MT} test set.

and test is the best configuration for the hybrid system. Extending the base lexicon for the runtime versions is useful for a hard integration but not for the soft one, and the same argument applies for multiple GF translations. Contrary to French, using a static lexicon is not always the best solution due to the limited amount of translation it gives. In our working test set, MAREC, the SIdev-SaEs system results to be better at syntactic level and for some representative lexical metrics such as TER and NIST. However, BLEU and METEOR still favour the static lexicon.

In summary, the hybrid systems presented in this section show improvements over the baseline which are moderate because of two reasons. First, SMT translations are already good for a start. And second, the amount of issues that GF handles are limited to be reflected on automatic metrics. The manual evaluation that is being carried out in WP9 will help to elucidate the strengths and weaknesses of the combination. For the moment, enlarging the static lexicons, specially for German, and extending the number of structures parsed by the grammar seem two crucial points to improve the architecture. As explained in Section 4.4, the system would also benefit from dealing with long distance agreement. In order to solve this issues and in parallel, a different architecture is being developed with the idea that a robust parsing of the input sentence and the long distance reorder and agreement come by construction.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
GF-Robust	82.09	64.70	81.25	7.69	2.51	19.76	23.71	16.21	42.33
GF-Sts	81.12	74.43	80.60	10.69	2.59	18.77	23.01	9.00	38.05
GF-SaBs	**70.34**	**61.00**	**69.24**	**19.63**	**3.78**	**26.35**	**35.55**	**20.73**	**68.39**
GF-SaEs	72.23	62.75	71.84	17.02	3.54	24.72	33.33	19.06	62.39
GF-UnBs	70.47	61.19	69.37	19.33	3.76	26.14	35.32	20.60	67.79
GF-UnEs	72.16	62.75	71.77	17.17	3.55	24.82	33.41	19.06	62.61

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
GF-Robust	19.32	14.20	12.75	14.20	2.49	73.22
GF-Sts	15.03	12.76	10.08	12.76	2.35	62.82
GF-SaBs	**24.00**	**21.07**	15.76	**21.07**	3.40	**99.19**
GF-SaEs	23.13	20.98	16.16	20.98	**3.44**	99.01
GF-UnBs	23.85	20.97	15.66	20.97	3.37	98.56
GF-UnEs	23.14	21.01	**16.25**	21.01	**3.44**	99.17

Table 28: Comparative table between the hybrid models based on GF for a subset of sentences of the MAREC test set.

7.2 Integration led by GF

GF has available a Resource Grammar Library and the general English lexicon which could serve as a general translator. But for this, one needs a robust parser to be able to deal with unknown structures not covered by the grammar and the lexicon. Such a parser has been described in Section 6. The probabilistic parser has been trained on the Penn Treebank and, in order to be applied to patents, also the lexicons developed as explained in Section 5 are used.

Including the extra lexicons does not solve one of the main issues with the current robust parser, that is, for efficiency issues, the length of the sentence to translate is limited to 25 tokens. Besides, the domain of the Penn Treebank is far from the biomedical one. So, at this point, we do not expect the robust statistical parsing to perform better than the standard GF parser applied on an in-domain GF grammar.

As a first experiment, we translate the MAREC test. The experiment has been conducted for the English-to-German translation. The whole test set cannot be used with the robust parser, only 537 out of the 1008 fragments fulfil the length restriction. These candidate sentences are then cleaned to fit the parser format. For instance, numbers are tokenised, punctuation removed and tags indicating the position of an image or a figure in the patent are also deleted. With this cleaning, the original sentence:

```
The use of claim 23 , wherein the amount of said composition is from 100
mg to 800 mg of ibuprofen .
```

would be converted into

```
the use of claim 2 3 wherein the amount of said composition is from 1 0 0
mg to 8 0 0 mg of ibuprofen
```

In the hybrid approach led by SMT, tokens such as punctuation and numbers are passed to the final SMT decoder. Here, since it is already known that GF is not dealing with them and they are preventing the parsing, the conflictive tokens are removed from the beginning.

Although the remaining 537 fragments can be parsed, not all of them can be properly linearised. This can be due to both structures not covered by the grammar and missing vocabulary. In this particular example, 98 sentences are finally fully linearised, a 20% of the candidate sentences. Notice that for Penn Treebank sentences, the percentage of success is a 7%. The repetitive structures of patents and the additional dictionary are probably the clue to the increment.

We evaluate this subset of sentences with the common set of metrics available for German (Table 28). According to all metrics, the standard GF translations with the lexicon built on runtime are better than the GF with robust parsing. Translations with robust parsing have a similar quality as those obtained with a static lexicon and the standard GF parser using the patents grammar. In the latter case, from the 614 chunks that `Genia` detects in the 98 sentences, 203 can be translated by GF and 411 are left in English in order to be evaluated as they are. 177 tokens correspond to punctuation and are not included in the chunks. So, using the same lexicons, the robust parsing is able to parse the full sentence but the necessary cleaning has damaged the final translation. The standard GF cannot translate all the terms of the sentence but keeps tokens such as numbers and punctuation that help to improve the performance. This is evident from the fact that metrics based on n-gram matching such as BLEU and NIST penalise the robust parser, whereas syntactic metrics for example favour this system.

8 Synergies and Relations

Some of the technology and resources developed in WP5 have been used in WP7 as the main case study. In fact, WP5 and WP7 share different corpora and tools. The corpora provided by EPO (see Section 2.2.2) is used to populate the patents retrieval system. The corpus is preprocessed using the in-house tokeniser developed for the patents domain. As a difference, WP7 deals with the original XML documents which must be preprocessed in the same way done to train the translation systems. This way the translation engines introduced in Sections 3, 4 and 7 can be integrated into the prototype. So, WP7 is in charge of extracting the raw text to translate from the original XML documents, preprocess it with our in-house tools maintaining the original mark-up, and post-process the translations to build the machine translated XML documents. The result of this pipeline is an XML document with the original text and the translations. The Patent MT and Retrieval Prototype can be accessed on-line at `http://molto-patents.ontotext.com`. See *D7.3 Patent MT and Retrieval. Final Report* [26] for a more detailed description.

This prototype has been build as a use case within the project, but patent translation is a hot topic nowadays and there exist several devoted projects, systems, applications and services. Google Patents[17] is a retrieval system for patents coming from the United States Patent and Trademark Office (USPTO). Patents issued in the United States are public domain documents, and images of the entire database of U.S. patents are readily available on-line via the USPTO website. Through this service, users can search 8 million patents and 3 million patent applications issued in the 1790s through the present, all of them in English.

Later, in February 2012 and in collaboration with EPO, Google Inc. launched Patent Translate. Patent Translate is an update to the Google Translate system that incorporates the EPO's parallel patent texts and allows translation between English and French, German, Spanish, Italian, Portuguese and Swedish. The translator is part of the EPO's Espacenet service[18] which provides free access to 70 million worldwide patent document. It can also be accessed through the European publication server[19]. The EPO plans to extend the service by the end of 2014 to permit users to translate from and to all 28 languages of the EPO members states, as well as provide for the translation of Chinese, Japanese, Korean, and Russian languages.

With these two collaborations, Google has added millions of patents to their training data, and the impact of this data on the quality of their translation system is significant. If one compares the tables given in this deliverable with the ones in D5.2, one can see the large improvement of their system from February 2011 to May 2012. Translations involving German have improved between 8 and 9 points of BLEU for example, whereas for the English-French translation the improvement is of 5 points.

A different approach is that of the Patent Language Translations Online (PLuTO) project. PLuTO is a dedicated project to patent translation, its MT framework is a web service whereby users can request translations. The translation engine uses the MaTrEx (Machine Translation Using Examples) system developed at DCU [4]. It is a hybrid data-driven system built following established design patterns, with an extensible framework allowing for the interchange of novel or previously developed modules as it is defined in [44]. The project is dealing with English, French, German, Spanish, Portuguese, Chinese and Japanese. For the languages we are interested in, they have at their disposal the MAREC corpus and EPO's translation memories (TM).

The data available to both Google and PLuTO makes the comparison with us difficult. We cannot use our MAREC's test set because it is included in PLuTO's training set. EPO's official data might be included in Google's corpus and reflected in the TM available to PLuTO. Since is difficult to control specially Google's data, we have also translated American patents with the mentioned systems. American patents are not translated to European languages, so there is no parallel corpus including these data. Monolingual data exist though, and it can be used for language modelling for example. MOLTO is not using

[17] https://www.google.com/?tbm=pts

[18] http://www.epo.org/searching/free/espacenet.html

[19] https://data.epo.org/publication-server

this data, and according to their publications neither does PLuTO. However, they might be part of Google's corpus. Still, we consider a test set build with these patents a fair comparison within translators.

The test set has been build from patents available trough Google Patents. We have retrieved 27 patents with IPC A61P and obtained the claims from the General Information section. With this, the UPSTO test set contains 1,000 fragments of English claims.

The fact that American patents are not translated implies that there are not references for the translations, so they cannot be evaluated with automatic metrics and we have to apply a human evaluation. This work is being carried out within WP9, Evaluation and will be reported in the *D9.2 MOLTO evaluation and assessment report.*

9 Summary

The work of WP5 has been devoted to improve the coverage and robustness within MOLTO. The main goal has been to explore the possibilities within the project framework to widen the GF coverage and robustness so that it approaches to translate open-domain text. Also the opposite approach has been investigated, that is, improving the translation of a general purpose engine (an SMT) with high quality translations given by GF.

The tasks related to widening GF have been focused on building general purpose lexicons. A more robust GF can be achieved by the use of the robust statistical parser that will allow to translate free text or, at least, the parts covered by the grammars without being affected by unknown elements. On the other hand, the development of a hybrid system between GF and SMT has been restricted to translate patents and has implied the construction of new resources on the domain and conceiving techniques to integrate both technologies.

Regarding the development of different types of general lexicons it has been used GF's core idea of common abstract syntax and multiple concrete syntaxes to produce multilingual morphological lexicons. The abstract syntax is based on data from the Princeton WordNet and the Oxford Advanced Learner's dictionary. The concrete syntaxes are produced using data from already existing lexical resources (i.e. Bilingual dictionaries and Universal WordNet), and GF's morphological smart paradigms. Because words can have multiple senses, and it is often very hard to find one-to-one word mappings between languages, two different types of multi-lingual lexicons have been developed: Uni-Sense and Multi-Sense. In a uni-sense lexicon each source word is restricted to represent one particular sense of the word, and hence it becomes easier to map it to one particular word in the target language. These type of lexicons are useful for building domain specific NLP applications. A multi-sense lexicon, on the other hand, is a more comprehensive lexicon and contains multiple senses of words and their translations to other languages. This type of lexicons can be used for open-domain tasks such as arbitrary text translation. These lexicons cover a number of language including English, German, Finnish, Bulgarian, Hindi, Urdu and their size ranges from 10 to 50 thousand lemmas.

In WP5 we also experimented with open-domain robust translation based solely on GF.

This is a huge step since the traditional application domain of GF is in controlled languages where the domain is small and well defined, while in the task of translating running text the source language is not clearly defined anymore. As a simple numerical measure for the leap, we can say that the typical GF applications deal with grammars containing hundreds of lemmas while in this experiment our grammars contain more than 50,000 lemmas. We developed an entirely new runtime system for GF in C which has the advantage to be more portable and more efficient. The efficiency was the first requirements that we had to satisfy since otherwise interpreting these huge grammars would be intractable. Furthermore, we turned the original non-probabilistic algorithms for parsing and reasoning into probabilistic ones. The introduction of probabilistic models is crucial for the disambiguation of the grammars which are by necessity highly ambiguous. The third major contribution to the project is that we also made the GF parser robust, i.e. when faced with sentences which are not parseable, it returns a sequence of recognized chunks rather than an error. We evaluated our implementation with state-of-the-art statistical parsers for related grammatical formalisms, and we found that for sentences longer than 25 tokens, our implementation is at least two orders of magnitude faster. We also tried to use our new architecture in machine translation but here the results are not conclusive yet. We found that the two main limitations are in the quality of the translation dictionaries which we built and the still limited coverage of the grammars. Furthermore, we need to better address the word sense disambiguation and the proper translation for multiword expressions.

The translation of patents using this robust parsing is still in an embryonic state, but we have developed a complete translation system that combines GF and SMT to overcome the input controlled language assumption. This hybrid system implies the construction of in-domain dictionaries and grammars that make use of probabilistic components, and the integration with an SMT engine that is able to complement GF translations. Regarding these resources for patent translation, we emphasise the generation of static lexicons obtained from SMT translation tables, and the on-line generation of lexicons with unseen vocabulary but available in the monolingual dictionaries. For German, also a dictionary of compounds has been built. A grammar for dealing with patents in English, French and German has been built on top of the resource grammar with several additions devoted to deal with chunks instead of sentences. Particular constructions appearing in patents are also covered by this new in-domain grammar. As a demand of the selected domain, we have also developed a detector and tokeniser of chemical compounds. A full translation system uses this tokeniser and prepares the patent to be translated. This involves chunking and parsing the source sentences which are first translated by GF and afterwards sent to an SMT decoder which is fed with this information. An SMT engine trained on the domain is also used by the top decoder. The final hybrid system is available for download and has several options that take into account which method to build the lexicon has to be used and which kind of integration is to be applied.

References

[1] Angelov, K. Incremental Parsing with Parallel Multiple Context-Free Grammars. In *European Chapter of the Association for Computational Linguistics* (2009).

[2] Angelov, K. *The Mechanics of the Grammatical Framework*. PhD thesis, Chalmers University of Technology, Gothenburg, Sweden, 2011.

[3] Angelov, K., and Ljunglöf, P. Fast Statistical Parsing with Parallel Multiple Context-Free Grammars. In *to appear in ACL* (2013).

[4] Armstrong, S., Flanagan, M., Graham, Y., Groves, D., Mellebeek, B., Morrissey, S., Stroppa, N., and Way, A. MaTrEx: Machine Translation Using Examples. In *TC-STAR OpenLab on Speech Translation* (Trento, Italy, 2006).

[5] Banerjee, S., and Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization* (2005).

[6] Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. The TIGER treebank. In *TLT'02, 1st Workshop on Treebanks and Linguistic Theories* (Sozopol, Bulgaria, 2002).

[7] de Melo, G., and Weikum, G. Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM conference on Information and knowledge management* (New York, NY, USA, 2009), CIKM '09, ACM, pp. 513–522.

[8] Détrez, G., and Ranta, A. Smart paradigms and the predictability and complexity of inflectional morphology. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2012), EACL '12, Association for Computational Linguistics, pp. 645–653.

[9] Doddington, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the 2nd Internation Conference on Human Language Technology* (2002), pp. 138–145.

[10] Enache, R., España-Bonet, C., Ranta, A., and Màrquez, L. A hybrid system for patent translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT12)* (Trento, Italy, may 2012), pp. 269–276.

[11] España-Bonet, C., Gonzàlez, M., and Màrquez, L. Description of the final collection of corpora. *MOLTO Deliverable 5.1*, D5.1 (09/2011 2011).

[12] España-Bonet, C., Màrquez, L., Enache, R., and Ranta, A. Description and evaluation of the combination prototypes. *MOLTO Deliverable 5.2*, D5.2 (03/2012 2012).

[13] Fellbaum, C. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[14] Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation 25*, 2 (2011), 127–144.

[15] Giménez, J., and Màrquez, L. Asiya: An Open Toolkit for Automatic Machine Translation (Meta-)Evaluation. *The Prague Bulletin of Mathematical Linguistics*, 94 (2010), 77–86.

[16] Giménez, J., and Màrquez, L. Linguistic features for automatic evaluation of heterogeneous mt systems. In *Proceedings of the Second ACL Workshop on Statistical Machine Translation* (June 2007), pp. 256–264.

[17] Giménez, J., and Màrquez, L. A smorgasbord of features for automatic mt evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation* (Columbus, Ohio, June 2008), The Association for Computational Linguistics, pp. 195–198.

[18] Jha, S., Narayan, D., Pande, P., and Bhattacharyya, P. A wordnet for hindi. In *International Workshop on Lexical Resources in Natural Language Processing* (2001).

[19] Kallmeyer, L., and Maier, W. Data-driven parsing with probabilistic linear context-free rewriting systems. In *Proceedings of the 23rd International Conference on Computational Linguistics* (Stroudsburg, PA, USA, 2010), COLING '10, Association for Computational Linguistics, pp. 537–545.

[20] Koehn, P., Hoang, H., Mayne, A. B., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computation Linguistics (ACL), Demonstration Session* (Jun 2007), pp. 177–180.

[21] Koehn, P., Shen, W., Federico, M., Bertoldi, N., Callison-Burch, C., Cowan, B., Dyer, C., Hoang, H., Bojar, O., Zens, R., Constantin, A., Herbst, E., and Moran, C. Open Source Toolkit for Statistical Machine Translation. Tech. rep., Johns Hopkins University Summer Workshop. http://www.statmt.org/jhuws/, 2006.

[22] Lin, C.-Y., and Och, F. J. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)* (2004).

[23] Lindén, K., and Carlson, L. FinnWordNet – WordNet på finska via översättning. *LexicoNordica – Nordic Journal of Lexicography 17* (2010), 119–140.

[24] Liu, D., and Gildea, D. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (Ann Arbor, Michigan, June 2005), Association for Computational Linguistics, pp. 25–32.

[25] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics 19* (June 1993), 313–330.

[26] Mateva, M., Gonzàlez, M., Enache, R., España-Bonet, C., Màrquez, L., Popov, B., and Ranta, A. Patent Case Study. Final Report. *MOLTO Deliverable 7.3*, D7.3 (03/2013 2013).

[27] Melamed, I. D., Green, R., and Turian, J. P. Precision and Recall of Machine Translation. In *Proceedings of the Joint Conference on Human Language Technology and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)* (2003).

[28] Mitton, R. A partial dictionary of English in computer-usable form. *Literary & Linguistic Computing 1*, 4 (Dec. 1986), 214–215.

[29] Niessen, S., Och, F. J., Leusch, G., and Ney, H. An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation* (2000).

[30] Och, F. J. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics* (Sapporo, Japan, July 6-7 2003).

[31] Och, F. J., and Ney, H. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (2002), pp. 295–302.

[32] Och, F. J., and Ney, H. A systematic comparison of various statistical alignment models. *Computational Linguistics 29*, 1 (2003), 19–51.

[33] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Association of Computational Linguistics* (2002), pp. 311–318.

[34] Popović, M., Stein, D., and Ney, H. Statistical machine translation of german compound words. In *FinTAL - 5th International Conference on Natural Language Processing, Springer Verlag, LNCS* (Turku, Finland, Aug. 2006), pp. 616–624.

[35] Ranta, A. The GF resource grammar library. *Linguistic Issues in Language Technology 2*, 1 (2009).

[36] Ranta, A. *Grammatical Framework: Programming with Multilingual Grammars.* CSLI Publications, Stanford, 2011. ISBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth).

[37] Romary, L., Salmon-Alt, S., and Francopoulo, G. Standards going concrete: from lmf to morphalou. In *Proceedings of the Workshop on Enhancing and Using Electronic Dictionaries* (Stroudsburg, PA, USA, 2004), ElectricDict '04, Association for Computational Linguistics, pp. 22–28.

[38] Seki, H., Matsumura, T., Fujii, M., and Kasami, T. On multiple context-free grammars. *Theoretical Computer Science 88*, 2 (October 1991), 191–229.

[39] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., , and Makhoul, J. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of AMTA* (2006), pp. 223–231.

[40] Stolcke, A. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics 21*, 2 (June 1995), 165–201.

[41] Stolcke, A. SRILM – An extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing* (2002).

[42] Täger, W. The Sentece-Aligned European Patent Corpus. *Proceedings of the 15th Annual Conference of the European Association for Machine Translation (EAMT 2011)* (2011).

[43] Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., and Sawaf, H. Accelerated DP based Search for Statistical Translation. In *Proceedings of European Conference on Speech Communication and Technology* (1997).

[44] Tinsley, J., Way, A., and Sheridan, P. PLuTO: MT for Online Patent Translation. In *Proceedings of the 9th Conferences of the Association for Machine Translation in the Americas (AMTA 2010)* (2010).

[45] Tsuruoka, Y., Tateishi, Y., Kim, J., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics.*, P. Bozanis and e. Houstis, E.N., Eds., vol. 3746. Springer Berlin Heidelberg, 2005, p. 382–392.

A Automatic Evaluation of the Hybrid Systems

This Appendix shows the complete automatic evaluation of the hybrid systems described in Section 7. Table 29 describes the characteristics of each of the systems here evaluated.

The following subsections report lexical and syntactic scores for the translation of two test sets, MAREC and EPO$_{\mathrm{MT}}$ (see Section 2.2 for the details).

A.1 Nomenclature

A.2 English-to-French translation

A.3 English-to-German translation

	Base lexicon	**Lexicon acquisition**	**Multiple GF options**	**Integration**	**GF on dev**
HI-Sts	–	static	○	hard	○
HI-SaBs	base	safe	○	hard	○
HI-UnBs	base	unsafe	○	hard	○
HI-SaEs	extended	safe	○	hard	○
HI-UnEs	extended	unsafe	○	hard	○
HI-Stm	–	static	●	hard	○
HI-SaBm	base	safe	●	hard	○
HI-UnBm	base	unsafe	●	hard	○
HI-SaEm	extended	safe	●	hard	○
HI-UnEm	extended	unsafe	●	hard	○
SI-Sts	–	static	○	soft	○
SI-SaBs	base	safe	○	soft	○
SI-UnBs	base	unsafe	○	soft	○
SI-SaEs	extended	safe	○	soft	○
SI-UnEs	extended	unsafe	○	soft	○
SI-Stm	–	static	●	soft	○
SI-SaBm	base	safe	●	soft	○
SI-UnBm	base	unsafe	●	soft	○
SI-SaEm	extended	safe	●	soft	○
SI-UnEm	extended	unsafe	●	soft	○
HIdev-Sts	–	static	○	hard	●
HIdev-SaBs	base	safe	○	hard	●
HIdev-UnBs	base	unsafe	○	hard	●
HIdev-SaEs	extended	safe	○	hard	●
HIdev-UnEs	extended	unsafe	○	hard	●
HIdev-Stm	–	static	●	hard	●
HIdev-SaBm	base	safe	●	hard	●
HIdev-UnBm	base	unsafe	●	hard	●
HIdev-SaEm	extended	safe	●	hard	●
HIdev-UnEm	extended	unsafe	●	hard	●
SIdev-Sts	–	static	○	soft	●
SIdev-SaBs	base	safe	○	soft	●
SIdev-UnBs	base	unsafe	○	soft	●
SIdev-SaEs	extended	safe	○	soft	●
SIdev-UnEs	extended	unsafe	○	soft	●
SIdev-Stm	–	static	●	soft	●
SIdev-SaBm	base	safe	●	soft	●
SIdev-UnBm	base	unsafe	●	soft	●
SIdev-SaEm	extended	safe	●	soft	●
SIdev-UnEm	extended	unsafe	●	soft	●

Table 29: Main characteristics of the hybrid systems analised in Section 7.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
HI-Sts	34.28	23.76	32.43	54.36	9.13	38.58	70.08	66.23	71.47
HIdev-Sts	**33.91**	**23.35**	**32.05**	54.79	**9.18**	38.91	70.44	**66.82**	**72.28**
HI-Stm	34.04	23.61	32.22	54.65	9.16	38.78	70.25	66.41	71.89
HIdev-Stm	33.99	23.66	32.20	**54.93**	9.17	**39.04**	**70.72**	66.79	72.20
HI-SaBs	44.95	33.27	42.64	43.73	7.80	32.44	61.71	52.94	51.77
HIdev-SaBs	44.04	31.84	41.75	44.55	7.89	32.98	61.78	53.69	53.46
HI-SaEs	44.48	32.80	42.14	44.27	7.88	32.87	62.31	55.30	53.14
HIdev-SaEs	44.00	31.98	41.71	44.77	7.92	33.18	62.33	55.44	54.01
HI-SaBm	39.10	27.80	37.09	48.65	8.41	35.06	64.62	57.48	61.24
HIdev-SaBm	38.47	26.94	36.47	49.22	8.48	35.44	64.81	58.19	62.43
HI-SaEm	38.53	27.17	36.54	49.48	8.53	35.74	65.51	60.23	63.03
HIdev-SaEm	38.02	26.36	36.07	50.12	8.59	36.18	65.66	60.43	64.04
HI-UnBs	44.75	32.94	42.39	43.88	7.82	32.52	61.91	53.17	52.20
HIdev-UnBs	43.65	31.12	41.32	44.87	7.93	33.13	61.85	53.92	54.20
HI-UnEs	44.30	32.50	41.92	44.36	7.90	32.90	62.48	55.53	53.50
HIdev-UnEs	43.47	31.01	41.21	45.11	7.98	33.36	62.34	55.99	54.98
HI-UnBm	38.97	27.51	36.93	48.79	8.43	35.14	64.77	57.67	61.58
HIdev-UnBm	38.31	26.54	36.27	49.34	8.50	35.51	64.94	58.51	62.84
HI-UnEm	38.43	26.94	36.41	49.55	8.54	35.73	65.62	60.40	63.27
HIdev-UnEm	37.84	26.05	35.84	50.23	8.61	36.19	65.81	60.80	64.44

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
HI-Sts	55.67	58.22	40.03	63.82	7.02	98.72
HIdev-Sts	**56.46**	**58.91**	**40.86**	**64.54**	7.06	**99.99**
HI-Stm	55.71	58.32	40.19	63.99	7.06	99.01
HIdev-Stm	56.02	58.81	40.36	64.36	**7.07**	99.51
HI-SaBs	47.79	50.72	35.63	55.54	6.19	86.33
HIdev-SaBs	48.84	51.91	36.40	56.38	6.31	88.06
HI-SaEs	48.10	50.82	35.56	56.06	6.21	86.64
HIdev-SaEs	48.67	51.51	36.24	56.43	6.28	87.74
HI-SaBm	50.84	54.11	37.67	59.14	6.64	91.94
HIdev-SaBm	51.66	55.01	38.06	59.79	6.72	93.16
HI-SaEm	51.45	54.59	37.99	59.87	6.68	92.81
HIdev-SaEm	52.13	55.18	38.55	60.22	6.77	93.89
HI-UnBs	48.16	50.95	35.91	55.87	6.21	86.81
HIdev-UnBs	49.21	51.94	36.66	56.75	6.35	88.58
HI-UnEs	48.35	51.01	35.81	56.37	6.22	87.05
HIdev-UnEs	49.09	51.91	36.62	57.02	6.33	88.52
HI-UnBm	51.13	54.31	37.98	59.45	6.66	92.40
HIdev-UnBm	52.12	55.26	38.36	60.20	6.74	93.73
HI-UnEm	51.62	54.75	38.15	60.12	6.70	93.13
HIdev-UnEm	52.33	55.45	38.75	60.61	6.79	94.32

Table 30: Hard integration hybrids, for the English-to-French translation of the MAREC test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
SI-Sts	27.45	18.43	25.88	62.30	9.94	44.99	75.90	72.44	61.95
SIdev-Sts	26.90	17.71	25.33	62.82	10.01	45.31	76.11	**73.63**	63.45
SI-Stm	27.44	18.41	25.86	62.32	9.94	45.01	75.91	72.46	61.99
SIdev-Stm	**26.81**	**17.69**	**25.23**	**63.05**	**10.03**	**45.57**	**76.21**	73.49	**63.69**
SI-SaBs	27.24	18.51	25.72	62.20	9.91	44.94	75.79	72.04	61.91
SIdev-SaBs	26.92	18.11	25.42	62.70	9.96	45.39	75.95	72.35	62.83
SI-SaEs	27.16	18.35	25.62	62.38	9.94	45.05	75.91	72.34	62.26
SIdev-SaEs	26.84	18.06	25.32	62.65	9.96	45.24	76.13	73.07	63.05
SI-SaBm	27.18	18.56	25.69	62.28	9.92	45.02	75.85	72.02	61.96
SIdev-SaBm	**26.71**	18.07	25.28	62.62	9.95	45.19	75.88	72.50	62.94
SI-SaEm	27.26	18.56	25.77	62.29	9.92	45.05	75.79	72.09	61.91
SIdev-SaEm	26.79	18.00	25.29	62.81	9.98	45.54	75.98	72.77	63.19
SI-UnBs	27.28	18.54	25.76	62.11	9.91	44.76	75.75	72.00	61.76
SIdev-UnBs	26.93	18.18	25.43	62.49	9.94	45.06	75.91	72.32	62.60
SI-UnEs	27.20	18.39	25.67	62.29	9.93	44.85	75.86	72.28	62.09
SIdev-UnEs	**26.71**	17.92	**25.20**	62.88	9.98	45.33	76.19	73.13	63.38
SI-UnBm	27.22	18.60	25.74	62.18	9.91	44.83	75.80	71.96	61.80
SIdev-UnBm	26.86	18.17	25.41	62.65	9.94	45.41	76.03	72.59	62.83
SI-UnEm	27.29	18.59	25.80	62.21	9.91	44.85	75.74	72.04	61.76
SIdev-UnEm	26.85	18.09	25.41	62.69	9.95	45.49	75.95	72.74	62.95

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
SI-Sts	62.60	65.06	48.62	47.02	3.57	97.89
SIdev-Sts	**64.05**	66.36	**50.25**	47.62	3.59	**99.74**
SI-Stm	62.60	65.06	48.62	47.02	3.57	97.89
SIdev-Stm	63.61	66.21	49.97	47.55	3.59	99.39
SI-SaBs	62.71	65.36	49.31	47.26	3.59	98.50
SIdev-SaBs	63.40	66.00	50.30	47.46	**3.61**	99.47
SI-SaEs	62.85	65.49	49.27	47.30	3.59	98.58
SIdev-SaEs	63.66	66.27	50.15	47.70	**3.61**	99.66
SI-SaBm	62.71	65.37	49.12	47.26	3.59	98.44
SIdev-SaBm	63.19	65.85	49.99	47.59	3.60	99.24
SI-SaEm	62.57	65.30	48.98	47.19	3.59	98.26
SIdev-SaEm	63.24	65.98	49.42	47.52	3.60	99.04
SI-UnBs	62.55	65.29	48.94	47.27	3.59	98.29
SIdev-UnBs	63.22	65.99	49.69	47.56	**3.61**	99.21
SI-UnEs	62.69	65.41	48.91	47.31	3.59	98.37
SIdev-UnEs	63.73	**66.49**	49.96	**47.74**	**3.61**	99.70
SI-UnBm	62.53	65.26	48.83	47.25	3.59	98.23
SIdev-UnBm	63.31	65.86	49.52	47.62	3.60	99.13
SI-UnEm	62.38	65.19	48.71	47.19	3.59	98.07
SIdev-UnEm	63.34	66.02	49.66	47.60	3.60	99.22

Table 31: Soft integration hybrids for the English-to-French translation of the MAREC test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
HI-Sts	37.22	23.90	35.22	53.38	9.15	33.96	65.90	61.62	66.64
HIdev-Sts	37.13	**23.71**	35.42	53.55	9.17	34.23	66.13	61.78	66.93
HI-Stm	**36.86**	23.73	**34.95**	**53.80**	**9.19**	34.30	**66.16**	**61.95**	**67.30**
HIdev-Stm	37.28	24.34	35.62	53.48	9.11	**34.36**	66.07	61.67	66.46
HI-SaBs	41.59	27.64	39.52	47.66	8.44	30.05	61.51	53.82	56.19
HIdev-SaBs	41.09	27.51	39.56	48.02	8.49	30.32	60.73	53.50	56.42
HI-SaEs	41.60	27.61	39.47	47.75	8.47	30.14	61.37	55.33	56.59
HIdev-SaEs	41.32	27.31	39.49	48.01	8.50	30.29	60.91	54.99	56.80
HI-SaBm	40.51	26.73	38.52	49.22	8.61	31.15	62.51	54.84	58.63
HIdev-SaBm	40.26	26.59	38.48	49.44	8.64	31.24	62.09	54.59	58.76
HI-SaEm	40.25	26.52	38.25	49.53	8.67	31.34	62.68	56.77	59.53
HIdev-SaEm	40.01	26.65	38.26	49.75	8.69	31.54	62.28	56.43	59.55
HI-UnBs	41.51	27.51	39.44	47.71	8.45	30.08	61.56	53.99	56.38
HIdev-UnBs	41.16	27.88	39.55	47.74	8.53	30.19	60.54	53.43	56.13
HI-UnEs	41.52	27.48	39.38	47.80	8.48	30.16	61.43	55.50	56.78
HIdev-UnEs	41.29	28.00	39.70	47.66	8.52	30.26	60.35	54.83	56.24
HI-UnBm	40.44	26.61	38.44	49.25	8.62	31.16	62.54	54.99	58.78
HIdev-UnBm	39.91	26.57	38.26	49.85	8.70	31.44	62.42	54.90	59.32
HI-UnEm	40.18	26.39	38.17	49.56	8.68	31.33	62.73	56.93	59.69
HIdev-UnEm	39.80	26.89	38.27	49.79	8.71	31.56	62.30	56.43	59.54

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
HI-Sts	53.54	58.07	40.21	66.15	7.40	99.56
HIdev-Sts	53.63	58.19	39.66	66.32	7.41	99.45
HI-Stm	**53.78**	**58.35**	**40.29**	**66.41**	**7.45**	**100.00**
HIdev-Stm	53.43	58.01	39.33	66.10	7.37	99.00
HI-SaBs	48.90	53.61	38.04	61.26	7.00	92.69
HIdev-SaBs	48.76	53.50	37.62	61.15	7.03	92.45
HI-SaEs	49.05	53.73	37.68	61.60	6.97	92.63
HIdev-SaEs	48.70	53.68	37.22	61.51	7.00	92.33
HI-SaBm	49.71	54.35	38.10	62.46	7.20	94.17
HIdev-SaBm	49.45	54.26	37.78	62.31	7.23	93.91
HI-SaEm	50.26	54.78	38.16	62.96	7.18	94.65
HIdev-SaEm	49.95	54.73	37.69	62.81	7.20	94.30
HI-UnBs	49.13	53.79	38.30	61.48	7.00	93.05
HIdev-UnBs	48.78	53.66	37.53	61.32	7.03	92.52
HI-UnEs	49.28	53.89	37.91	61.83	6.97	92.96
HIdev-UnEs	48.81	53.49	37.26	61.49	7.00	92.30
HI-UnBm	49.95	54.51	38.42	62.67	7.20	94.55
HIdev-UnBm	49.90	54.60	38.37	62.67	7.26	94.69
HI-UnEm	50.48	54.95	38.44	63.19	7.18	95.00
HIdev-UnEm	50.15	54.87	38.27	62.90	7.23	94.82

Table 32: Hard integration hybrids, for the English-to-French translation of the EPO$_{\mathrm{MT}}$ test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
SI-Sts	30.14	19.02	28.38	61.11	9.94	40.72	72.44	70.16	63.76
SIdev-Sts	**29.95**	19.26	28.51	**61.28**	**9.99**	**40.94**	72.43	**70.66**	**63.90**
SI-Stm	30.13	**18.99**	**28.37**	61.13	9.94	40.74	**72.46**	70.16	63.80
SIdev-Stm	30.14	19.26	28.67	61.15	9.96	40.91	72.25	70.48	63.60
SI-SaBs	30.42	19.37	28.72	60.63	9.87	40.43	72.06	69.44	62.82
SIdev-SaBs	30.58	19.84	29.19	60.50	9.86	40.29	71.66	68.95	62.04
SI-SaEs	30.28	19.23	28.55	60.87	9.90	40.59	72.28	69.79	63.28
SIdev-SaEs	30.66	19.65	29.12	60.14	9.82	40.14	71.58	69.42	62.05
SI-SaBm	30.41	19.35	28.71	60.65	9.87	40.41	72.09	69.38	62.83
SIdev-SaBm	30.71	19.88	29.17	60.30	9.83	40.28	71.32	69.20	61.87
SI-SaEm	30.24	19.16	28.51	60.91	9.91	40.61	72.37	69.81	63.39
SIdev-SaEm	30.25	19.52	28.59	60.97	9.91	40.75	72.01	69.47	63.07
SI-UnBs	30.45	19.40	28.75	60.58	9.86	40.39	72.02	69.33	62.72
SIdev-UnBs	30.72	19.71	29.18	60.32	9.83	40.33	71.63	69.14	62.03
SI-UnEs	30.27	19.24	28.55	60.85	9.90	40.57	72.26	69.77	63.25
SIdev-UnEs	30.42	19.57	28.75	60.74	9.88	40.59	72.01	69.27	62.73
SI-UnBm	30.44	19.38	28.75	60.60	9.86	40.36	72.05	69.26	62.72
SIdev-UnBm	30.92	20.25	29.45	60.17	9.80	40.12	71.02	68.34	61.12
SI-UnEm	30.23	19.17	28.51	60.90	9.90	40.60	72.35	69.80	63.37
SIdev-UnEm	30.54	20.03	29.13	60.44	9.86	40.43	71.32	68.90	61.91

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
SI-StEs	62.67	66.50	52.03	73.76	8.00	99.48
SIdev-StEs	62.91	**67.21**	52.10	**73.90**	**8.02**	**99.90**
SI-StEm	62.68	66.55	51.98	73.76	7.99	99.47
SIdev-StEm	**62.96**	67.04	**52.23**	73.87	8.00	99.85
SI-SaBs	62.50	66.31	52.13	73.34	7.95	99.19
SIdev-SaBs	62.14	66.17	51.96	72.91	7.96	98.86
SI-SaEs	62.56	66.36	51.97	73.58	7.98	99.28
SIdev-SaEs	62.21	66.28	51.92	73.14	7.92	98.86
SI-SaBm	62.41	66.25	51.93	73.29	7.96	99.06
SIdev-SaBm	62.15	66.24	51.95	73.05	7.95	98.90
SI-SaEm	62.57	66.42	51.89	73.60	7.98	99.29
SIdev-SaEm	62.63	66.43	52.11	73.37	8.00	99.39
SI-UnBs	62.40	66.22	52.06	73.26	7.95	99.07
SIdev-UnBs	62.23	66.14	51.84	72.99	7.94	98.81
SI-UnEs	62.52	66.33	51.95	73.56	7.98	99.25
SIdev-UnEs	62.31	66.29	51.86	73.21	7.97	99.04
SI-UnBm	62.32	66.16	51.86	73.21	7.95	98.95
SIdev-UnBm	62.08	66.04	52.23	72.53	7.92	98.72
SI-UnEm	62.56	66.40	51.89	73.58	7.98	99.27
SIdev-UnEm	62.22	66.30	51.81	72.93	7.96	98.88

Table 33: Soft integration hybrids for the English-to-French translation of the EPO$_{\mathrm{MT}}$ test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
HI-StBs	36.75	26.25	34.89	52.91	8.77	38.44	64.89	58.72	73.07
HIdev-StBs	35.93	**24.53**	34.17	53.33	8.91	38.74	64.61	**59.31**	74.47
HI-StEm	36.62	26.07	34.72	53.00	8.79	38.48	**64.95**	58.87	73.30
HIdev-StEm	**35.91**	24.60	**34.03**	**53.36**	**8.92**	**38.75**	64.56	59.26	**74.48**
HI-SaBs	48.13	36.04	45.92	43.23	7.62	32.05	56.60	48.33	54.61
HIdev-SaBs	47.53	34.88	45.47	43.47	7.69	32.10	56.00	48.46	55.32
HI-SaEs	50.61	39.13	48.79	41.12	7.25	30.75	54.18	44.05	49.49
HIdev-SaEs	50.35	38.35	48.47	41.19	7.30	30.81	53.78	44.04	49.90
HI-SaBm	47.05	35.04	44.73	44.17	7.77	32.65	58.12	50.18	56.80
HIdev-SaBm	46.76	34.01	44.55	44.34	7.80	32.70	57.20	49.97	57.13
HI-SaEm	48.84	37.29	46.89	42.47	7.48	31.56	56.06	46.56	52.78
HIdev-SaEm	48.53	36.43	46.52	42.68	7.54	31.71	55.76	46.70	53.38
HI-UnBs	48.21	36.13	46.01	43.16	7.61	32.01	56.62	48.22	54.48
HIdev-UnBs	47.53	34.85	45.42	43.54	7.69	32.17	56.17	48.45	55.43
HI-UnEs	50.72	39.23	48.90	40.97	7.24	30.66	54.18	43.92	49.29
HIdev-UnEs	50.35	38.30	48.50	41.20	7.30	30.79	53.84	44.05	49.92
HI-UnBm	47.07	35.06	44.75	44.19	7.77	32.67	58.17	50.16	56.81
HIdev-UnBm	46.69	33.93	44.43	44.46	7.81	32.75	57.45	50.01	57.32
HI-UnEm	49.00	37.43	47.05	42.32	7.47	31.47	55.98	46.35	52.51
HIdev-UnEm	48.65	36.43	46.63	42.52	7.53	31.59	55.61	46.59	53.17

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
HI-StBs	48.45	45.33	27.23	45.33	6.30	98.30
HIdev-StBs	48.76	**45.94**	27.43	**45.94**	6.39	99.37
HI-StEm	48.60	45.43	27.34	45.43	6.32	98.59
HIdev-StEm	**49.10**	45.93	**27.97**	45.93	**6.42**	**100.00**
HI-SaBs	40.74	39.32	22.47	39.32	5.65	84.51
HIdev-SaBs	41.05	39.66	22.99	39.66	5.73	85.54
HI-SaEs	38.36	37.80	22.60	37.80	5.73	82.56
HIdev-SaEs	38.44	38.05	22.78	38.05	5.77	83.06
HI-SaBm	41.78	40.03	22.92	40.03	5.69	86.01
HIdev-SaBm	41.92	40.13	23.16	40.13	5.71	86.37
HI-SaEm	39.79	38.97	22.79	38.97	5.79	84.47
HIdev-SaEm	39.98	39.32	23.04	39.32	5.82	85.12
HI-UnBs	40.68	39.25	22.46	39.25	5.65	84.40
HIdev-UnBs	41.08	39.62	23.05	39.62	5.72	85.55
HI-UnEs	38.24	37.69	22.57	37.69	5.73	82.37
HIdev-UnEs	38.37	38.13	22.94	38.13	5.78	83.24
HI-UnBm	41.80	40.01	22.94	40.01	5.69	86.01
HIdev-UnBm	42.20	40.30	23.40	40.30	5.72	86.84
HI-UnEm	39.67	38.76	22.71	38.76	5.78	84.15
HIdev-UnEm	39.93	39.25	23.06	39.25	5.81	85.03

Table 34: Hard integration hybrids, for the English-to-German translation of the MAREC test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
SI-Sts	31.87	**22.92**	30.21	57.43	9.35	42.88	67.47	62.55	62.58
SIdev-Sts	31.06	22.95	29.45	**57.70**	9.44	43.14	**67.73**	63.22	**63.61**
SI-Stm	31.89	**22.92**	30.22	57.42	9.35	42.86	67.47	62.54	62.56
SIdev-Stm	31.59	22.97	29.97	57.35	9.41	42.82	67.43	62.91	62.86
SI-SaBs	32.05	23.00	30.36	57.13	9.35	42.72	67.23	62.90	62.31
SIdev-SaBs	31.03	23.49	29.49	57.22	9.46	43.21	67.15	63.28	63.17
SI-SaEs	31.62	22.97	29.97	57.45	9.38	43.09	67.50	63.02	62.94
SIdev-SaEs	**30.80**	23.39	**29.24**	57.40	**9.48**	43.38	67.31	**63.47**	63.59
SI-SaBm	32.21	22.95	30.46	57.20	9.33	42.73	67.33	62.82	62.23
SIdev-SaBm	31.54	23.66	29.99	56.69	9.38	42.75	66.82	62.99	62.21
SI-SaEm	31.91	23.01	30.22	57.32	9.35	42.93	67.44	62.74	62.53
SIdev-SaEm	31.08	23.43	29.55	57.18	9.45	43.20	67.18	63.27	63.15
SI-UnBs	32.16	23.07	30.45	57.00	9.33	42.55	67.20	62.73	62.06
SIdev-UnBs	31.34	23.57	29.71	56.81	9.43	42.74	66.90	63.06	62.56
SI-UnEs	31.65	23.01	29.99	57.39	9.37	43.01	67.52	62.97	62.85
SIdev-UnEs	30.82	23.54	29.29	57.32	**9.48**	**43.42**	67.21	63.34	63.43
SI-UnBm	32.29	23.00	30.54	57.08	9.31	42.58	67.32	62.69	62.02
SIdev-UnBm	31.39	23.60	29.76	56.77	9.43	42.80	66.90	63.10	62.52
SI-UnEm	31.94	23.05	30.25	57.24	9.34	42.83	67.45	62.68	62.42
SIdev-UnEm	31.54	23.63	29.98	56.72	9.38	42.78	66.85	62.61	62.17

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
SI-Sts	52.63	50.58	32.38	50.58	6.83	98.47
SIdev-Sts	52.98	50.83	32.62	50.83	6.89	99.11
SI-Stm	52.62	50.59	32.38	50.59	6.83	98.47
SIdev-Stm	52.81	50.71	32.39	50.71	6.89	98.81
SI-SaBs	52.62	50.71	32.53	50.71	6.82	98.62
SIdev-SaBs	52.71	51.02	32.88	51.02	6.94	99.47
SI-SaEs	52.64	50.87	32.75	50.87	6.86	99.02
SIdev-SaEs	**53.04**	51.13	33.16	51.13	**6.95**	**99.87**
SI-SaBm	52.62	50.75	32.61	50.75	6.81	98.70
SIdev-SaBm	52.61	50.95	32.79	50.95	6.86	99.09
SI-SaEm	52.51	50.92	32.77	50.92	6.87	99.03
SIdev-SaEm	52.77	51.01	32.64	51.01	6.92	99.27
SI-UnBs	52.45	50.62	32.33	50.62	6.80	98.32
SIdev-UnBs	52.73	50.93	32.75	50.93	6.91	99.25
SI-UnEs	52.59	50.84	32.68	50.84	6.85	98.91
SIdev-UnEs	52.92	51.01	33.01	51.01	**6.95**	99.64
SI-UnBm	52.47	50.67	32.43	50.67	6.79	98.41
SIdev-UnBm	52.77	50.95	32.75	50.95	6.90	99.23
SI-UnEm	52.45	50.88	32.70	50.88	6.85	98.89
SIdev-UnEm	52.65	**51.26**	**33.19**	**51.26**	6.91	99.73

Table 35: Soft integration hybrids for the English-to-German translation of the MAREC test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
HI-Sts	39.85	27.12	37.88	54.24	8.97	33.19	60.39	55.47	71.24
HIdev-Sts	39.29	27.97	37.31	53.93	9.08	33.20	59.62	55.70	71.21
HI-Stm	39.79	**27.08**	37.75	**54.29**	8.98	33.20	**60.43**	55.53	71.35
HIdev-Stm	**39.23**	27.96	**37.20**	53.99	**9.10**	**33.36**	59.74	**55.79**	**71.41**
HI-SaBs	48.15	34.24	46.12	46.67	8.02	28.22	54.13	46.37	56.49
HIdev-SaBs	47.30	33.12	45.36	47.40	8.12	28.53	53.78	46.37	57.63
HI-SaEs	51.13	37.27	49.25	44.77	7.70	27.01	51.19	42.37	51.12
HIdev-SaEs	50.50	36.12	48.64	45.37	7.78	27.24	50.96	42.60	52.16
HI-SaBm	47.48	33.51	45.35	47.06	8.11	28.46	54.95	47.45	57.81
HIdev-SaBm	47.02	32.62	45.06	47.46	8.16	28.63	53.98	46.94	58.20
HI-SaEm	49.58	35.68	47.53	45.70	7.87	27.58	52.66	44.37	53.89
HIdev-SaEm	48.91	34.53	46.90	46.35	7.96	27.86	52.57	44.70	55.03
HI-UnBs	48.20	34.28	46.17	46.60	8.01	28.16	54.09	46.23	56.37
HIdev-UnBs	47.40	33.19	45.44	47.27	8.12	28.42	53.68	46.25	57.43
HI-UnEs	51.18	37.31	49.29	44.74	7.69	26.97	51.18	42.23	51.02
HIdev-UnEs	50.57	36.24	48.70	45.41	7.78	27.23	51.03	42.46	52.08
HI-UnBm	47.48	33.52	45.36	47.04	8.10	28.44	54.93	47.43	57.78
HIdev-UnBm	47.03	32.60	45.09	47.40	8.15	28.58	53.99	46.99	58.18
HI-UnEm	49.58	35.68	47.53	45.68	7.87	27.56	52.72	44.35	53.88
HIdev-UnEm	48.77	34.38	46.77	46.49	7.98	27.93	52.65	44.71	55.25

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
HI-StBs	**46.32**	42.66	**24.74**	63.27	6.06	99.66
HIdev-StBs	46.25	42.18	24.49	63.12	6.12	99.38
HI-StEm	**46.32**	**42.69**	24.64	63.26	6.06	99.60
HIdev-StEm	46.25	42.49	24.51	**63.39**	**6.14**	**99.67**
HI-SaBs	39.25	37.09	21.00	60.81	5.55	88.53
HIdev-SaBs	39.33	37.03	21.15	60.77	5.64	88.95
HI-SaEs	36.76	35.56	20.61	60.90	5.55	86.47
HIdev-SaEs	36.78	35.36	20.60	60.66	5.55	86.31
HI-SaBm	39.97	37.67	21.34	61.21	5.57	89.61
HIdev-SaBm	39.82	37.51	21.55	60.95	5.67	89.85
HI-SaEm	38.09	36.74	20.94	61.16	5.59	88.07
HIdev-SaEm	38.21	36.57	20.76	60.92	5.57	87.75
HI-UnBs	39.17	36.99	20.96	60.68	5.54	88.36
HIdev-UnBs	39.29	36.91	21.11	60.58	5.63	88.75
HI-UnEs	36.65	35.46	20.57	60.82	5.55	86.31
HIdev-UnEs	36.61	35.28	20.52	60.46	5.53	86.01
HI-UnBm	39.97	37.67	21.33	61.21	5.57	89.60
HIdev-UnBm	39.81	37.52	21.40	61.00	5.64	89.68
HI-UnEm	38.05	36.67	21.05	61.14	5.59	88.10
HIdev-UnEm	38.25	36.57	20.93	60.94	5.58	87.95

Table 36: Hard integration hybrids, for the English-to-German translation of the $\mathrm{EPO}_{\mathrm{MT}}$ test set.

	WER	PER	TER	BLEU	NIST	GTM-2	MTR-st	RG-S*	ULC
SI-StBs	35.50	25.36	33.64	58.60	9.49	37.84	63.55	61.13	63.04
SIdev-StBs	**34.76**	25.56	**32.92**	**58.93**	9.59	**38.33**	**63.74**	61.31	**63.89**
SI-StEm	35.50	25.36	33.65	58.60	9.49	37.83	63.55	61.11	63.02
SIdev-StEm	35.17	**25.28**	33.22	58.55	9.56	37.88	63.59	**61.56**	63.52
SI-SaBs	35.91	25.66	34.09	58.11	9.43	37.46	63.16	60.95	62.17
SIdev-SaBs	35.38	26.78	33.55	57.56	9.55	37.98	62.34	61.06	62.10
SI-SaEs	35.59	25.64	33.73	58.22	9.47	37.61	63.37	61.23	62.63
SIdev-SaEs	34.86	26.22	32.94	58.13	**9.60**	38.31	62.98	61.44	63.26
SI-SaBm	35.93	25.68	34.14	58.07	9.43	37.38	63.15	60.87	62.07
SIdev-SaBm	35.42	26.54	33.65	57.63	9.52	37.84	62.42	60.79	62.04
SI-SaEm	35.56	25.63	33.74	58.23	9.47	37.60	63.38	61.19	62.64
SIdev-SaEm	34.98	26.30	33.10	57.78	9.58	37.90	62.79	61.23	62.80
SI-UnBs	35.93	25.65	34.11	58.08	9.43	37.44	63.18	60.92	62.14
SIdev-UnBs	35.03	26.24	33.23	57.96	9.57	37.94	62.82	61.23	62.82
SI-UnEs	35.58	25.62	33.72	58.23	9.47	37.62	63.39	61.27	62.67
SIdev-UnEs	35.24	26.76	33.45	57.53	9.54	37.97	62.37	61.05	62.18
SI-UnBm	35.97	25.69	34.18	58.03	9.42	37.36	63.15	60.82	62.01
SIdev-UnBm	34.98	26.31	33.21	57.96	9.57	37.99	62.83	61.15	62.81
SI-UnEm	35.55	25.61	33.73	58.25	9.47	37.61	63.41	61.23	62.68
SIdev-UnEm	35.40	26.42	33.49	57.54	9.51	37.66	62.58	60.95	62.13

	CP-Oc(*)	CP-Op(*)	CP-STM-9	SP-Op(*)	SP-pNIST-5	ULC
SI-StBs	51.92	50.01	32.41	68.08	6.66	98.74
SIdev-StBs	**52.32**	**50.46**	**33.16**	**68.76**	**6.75**	**100.00**
SI-StEm	51.92	50.00	32.41	68.12	6.66	98.75
SIdev-StEm	52.04	49.92	32.15	68.21	6.69	98.72
SI-SaBs	51.53	49.47	31.96	67.59	6.57	97.69
SIdev-SaBs	51.97	49.50	32.44	67.60	6.73	98.63
SI-SaEs	51.95	49.88	32.44	67.87	6.59	98.45
SIdev-SaEs	52.13	50.06	32.69	68.23	**6.75**	99.33
SI-SaBm	51.54	49.44	31.96	67.64	6.56	97.67
SIdev-SaBm	51.57	49.51	32.15	67.82	6.69	98.26
SI-SaEm	51.81	49.76	32.25	67.74	6.57	98.13
SIdev-SaEm	51.83	49.73	32.12	67.91	6.71	98.54
SI-UnBs	51.66	49.50	32.01	67.60	6.57	97.79
SIdev-UnBs	51.70	49.46	32.10	67.63	6.70	98.24
SI-UnEs	51.96	49.92	32.41	67.89	6.59	98.46
SIdev-UnEs	52.25	49.82	32.68	67.69	6.74	99.09
SI-UnBm	51.65	49.46	32.03	67.66	6.57	97.79
SIdev-UnBm	51.82	49.40	32.13	67.69	6.70	98.30
SI-UnEm	51.83	49.81	32.26	67.78	6.57	98.19
SIdev-UnEm	51.62	49.63	32.25	67.95	6.68	98.40

Table 37: Soft integration hybrids for the English-to-German translation of the EPO$_{\mathrm{MT}}$ test set.

B One-click System

B.1 Installation

The hybrid systems developed in this workpackage have been build using freely available software and in-house components. The system can be used in a regular machine with the previous installation of standard translation software: Moses, SRILM and GF. For this gcc and Haskell are needed. The specific GF grammar for patents and the Genia chunker are needed to deal with the biomedical domain. Perl is also necessary in order to run the main script and the tokeniser for the biomedical domain. In the following, we list the components together with information about where to download every component and how to install it.

- *Main script*
  Download: From consortium
  Install: No need; modify the script with the correct paths to the other software

- *In-domain tokeniser*
  Download: From consortium
  Install: No need

- *Genia chunker* [45]
  Download: http://www.nactem.ac.uk/GENIA/tagger/
  Install: Own README file

- *Grammatical Framework* [36]
  Download: http://www.grammaticalframework.org/download/index.html
  Install: Instructions in the same download link

- *GF patents grammar*
  Download: From consortium
  Install: Compile the grammar modules with ghc

  ```
  ghc --make -O3 MainTranslate
  ```

- *SRILM* [41]
  Download: http://www.speech.sri.com/projects/srilm/download.html
  Install: Own INSTALL file

- *Moses* [20]
  Download: git clone git://github.com/moses-smt/mosesdecoder.git
  Install: Own BUILD-INSTRUCTIONS.txt file

With these components working, running the main script with the file to translate and the language pair is enough. Running the script without this information displays a help screen:

```
csmisc14:hybrid cristina$ perl H1PTrad.pl

Usage: perl H1PTrad.pl -v # -m [runtime|unsafe|demo] <input> [src2trg]
-v: verbosity [0,1,2]
-m: mode [runtime|unsafe|demo]
input: file to translate
src2trg: language pair

Ex: perl H1PTrad.pl -v 1 -m demo /Users/cristina/hybrid/input/patsA61P.test.en en2fr
```

The first parameter indicates the amount of verbosity desired during the execution. The command "`-v 0`" displays no information at all, "`-v 1`" prints information about the ongoing step and the time it takes. Finally, "`-v 2`" also displays the output of the `Moses` decoder in the console.

Next, "`-m`" allows to choose the characteristics of the GF system. "`-m demo`" uses a static lexicon, it is both the fastest version and the one with the best performance in our experiments. "`-m runtime`" and "`-m unsafe`" build the lexicon at runtime in safe and unsafe modes (see Section 4 for the exact definion). By default, the translator uses the extended base lexicon and generates multiple GF translations.

The behaviour of the top SMT system is fixed within the script. It uses a soft integration of GF and the base SMT translator, and the weights fitted in the development process are obtained when the GF translations are also available. This default configuration corresponds to the SIdev-Stm system.

The third parameter "`<input>`" corresponds to the absolut path to the file to be translated. This file must be a raw text, one fragment per line, in the same way of `Moses` input files. Text should not be tokenised and the original capitalisation should be kept.

Finally, "`[src2trg]`" indicates the source language (src) and the target language one wants to translate into (trg). "`en`", "`fr`" and "`de`" should be used for English, French and German respectively.