Learning Domain-Specific Transfer Rules: An Experiment with Korean to English Translation

Benoit Lavoie, Michael White, and Tanya Korelsky
CoGenTex, Inc.
840 Hanshaw Road
Ithaca, NY 14850, USA
benoit,mike,tanya@cogentex.com

Abstract

We describe the design of an MT system that employs transfer rules induced from parsed bitexts and present evaluation results. The system learns lexico-structural transfer rules using syntactic pattern matching, statistical co-occurrence and error-driven filtering. In an experiment with domain-specific Korean to English translation, the approach yielded substantial improvements over three baseline systems.

1 Introduction

In this paper, we describe the design of an MT system that employs transfer rules induced from parsed bitexts and present evaluation results for Korean to English translation. Our approach is based on lexico-structural transfer (Nasr et. al., 1997), and extends recent work reported in (Han et al., 2000) about Korean to English transfer in particular. Whereas Han et al. focus on high quality domain-specific translation using handcrafted transfer rules, in this work we instead focus on automating the acquisition of such rules.

The proposed approach is inspired by example-based machine translation (EBMT; Nagao, 1984; Sato and Nagao, 1990; Maruyama and Watanabe, 1992) and is similar to the recent works of (Meyers et al., 1998) and (Richardson et al., 2001) where transfer rules are also derived after aligning the source and target nodes of corresponding parses. However, while (Meyers et al., 1998) and (Richardson et al., 2001) only consider parses and rules with lexical labels and syntactic roles, our approach uses parses containing any syntactic information provided by parsers (lexical labels, syntactic roles, tense, number, person, etc.), and derives rules consisting of any source and target tree sub-patterns matching a subset of the parse features. A more detailed description of the differences can be found in (Lavoie et. al., 2001).

2 Overall Runtime System Design

Our Korean to English MT runtime system relies on the following off-the-shelf software components:

Korean parser For parsing, we used the wide coverage syntactic dependency parser for Korean developed by (Yoon et al., 1997). The parser was not trained on our corpus.

Transfer component For transfer of the Korean parses to English structures, we used the same lexico-structural transfer framework as (Lavoie et al., 2000).

Realizer For surface realization of the transferred English syntactic structures, we used the RealPro English realizer (Lavoie and Rambow, 1997).

The training of the system is described in the next two sections.

3 Data Preparation

3.1 Parses for the Bitexts

In our experiments, we used a parallel corpus derived from bilingual training manuals provided by the U.S. Defense Language Institute. The corpus consists of a Korean dialog of 4,183 sentences about battle scenario message traffic and their English human translations.

The parses for the Korean sentences were obtained using Yoon's parser, as in the runtime system. The parses for the English human transla-

```
(S1) {i}    {Ci-To-Reul}     {Ta-Si}  {Po-Ra}.
     this + map-accusative + again +  look-imp
(D1) {po} [class=vbma ente={ra}] (
       s1 {ci-to} [class=nnin2 ppca={reul}] (
         s1 {i} [class=ande] )
       s1 {ta-si} [class=adco2] )
(S2) Look at the map again.
(D2) look [class=verb mood=imp] (
       attr at [class=preposition] (
         ii map [class=common_noun article=def] )
       attr again [class=adverb] )
```

Figure 1: Syntactic dependency representations for corresponding Korean and English sentences

	Korean	English
Avg. sentence size	9.08	13.26
Avg. parse size	8.96	10.77

Table 1: Average sizes for sentences and parses in corpus

tions were derived from an English Tree Bank developed in (Han et al., 2000). To enable the surface realization of the English parses via RealPro, we automatically converted the phrase structures of the English Tree Bank into deep-syntactic dependency structures (DSyntSs) of the Meaning-Text Theory (MTT) (Mel'čuk, 1988) using Xia's converter (Xia and Palmer, 2001) and our own conversion grammars. The realization results of the resulting DSyntSs for our training corpus yielded a unigram and bigram accuracy (f-score) of approximately 95% and 90%, respectively.

A DSyntS is an *unordered tree* where all nodes are *meaning-bearing* and *lexicalized*. Since the output of the Yoon parser is quite similar, we have used its output as is. The syntactic dependency representations for two corresponding Korean[1] and English sentences are shown in Figure 1.

3.2 Training and Test Sets of Parse Pairs

The average sentence lengths (in words) and parse sizes (in nodes) for the 4,183 Korean and English sentences in our corpus are given in Table 1.

In examining the Korean parses, we found that many of the larger parses, especially those containing intra-sentential punctuation, had incorrect dependency assignments, incomplete lemmatization or were incomplete parses. In examining the English converted parses, we found that many of the parses containing intra-sentential punctuation marks other than commas had incorrect dependency assignments, due to the limitations of our conversion grammars. Consequently, in our experiments we have primarily focused on a higher quality subset of 1,483 sentence pairs, automatically selected by eliminating from the corpus all parse pairs where one of the parses contained more than 11 content nodes or involved problematic intra-sentential punctuation.

	Korean	English
Avg. sentence size	6.43	9.36
Avg. parse size	6.43	7.34

Table 2: Average sizes for sentences and parses in training set

	Korean	English
Avg. sentence size	7.04	10.58
Avg. parse size	7.02	8.56

Table 3: Average sizes for sentences and parses in test set

We divided this higher quality subset into training and test sets. For the test set, we randomly selected 50 parse pairs containing at least 5 nodes each. For the training set, we used the remaining 1,433 parse pairs. The average sentence lengths and parse sizes for the training and test sets are represented in Tables 2 and 3.

3.3 Creating the Baseline Transfer Dictionary

In our system, transfer dictionaries contain Korean to English lexico-structural transfer rules defined using the formalism described in (Nasr et. al., 1997), extended to include log likelihood ratios (Manning and Schutze, 1999: 172-175). Sample transfer rules are illustrated in Section 4. The simplest transfer rules consist of direct lexical mappings, while the most complex may contain source and target syntactic patterns composed of multiple nodes defined with lexical and/or syntactic features. Each transfer rule is assigned a log likelihood ratio calculated using the training parse set.

To create the baseline transfer dictionary for our experiments, we had three bilingual dictionary resources at our disposal:

A corpus-based handcrafted dictionary: This dictionary was manually assembled by (Han et al., 2000) for the same corpus used here. Note,

[1] Korean is represented in romanized format in this paper.

	Korean	English
Lexical coverage	92.18%	90.17%

Table 4: Concurrent lexical coverage of training set by baseline dictionary

however, that it was developed for different parse representations, and with an emphasis primarily on the lexical coverage of the source parses, rather than the source and target parse pairs.

A corpus-based extracted dictionary: This dictionary was automatically created from our corpus by the RALI group from the University of Montreal. Since the extraction heuristics did not handle the rich morphological suffixes of Korean, the extraction results contained inflected words rather than lexemes.

A wide coverage dictionary: This dictionary of 70,300 entries was created by Systran, without regard to our corpus.

We processed and combined these resources as follows:

- First, we replaced the inflected words with uninflected lexemes using Yoon's morphological analyzer and a wide coverage English morphological database (Karp and Schabes, 1992).
- Second, we merged all morphologically analyzed entries after removing all non-lexical features, since these features generally did not match those found in the parses.
- Third, we matched the resulting transfer dictionary entries with the training parse set, in order to determine for each entry all possible part-of-speech instantiations and dependency relationships. For each distinct instantiation, we calculated a log likelihood ratio.
- Finally, we created a baseline dictionary using the instantiated rules whose source patterns had the best log likelihood ratios.

Table 4 illustrates the concurrent lexical coverage of the training set using the resulting baseline dictionary, i.e. the percentage of nodes covered by rules whose source and target patterns both match. Note that since the baseline dictionary contained some noise, we allowed induced rules to override ones in the baseline dictionary where applicable.

```
@KOREAN:
    {po} [class=vbma] (
      s1 $X [ppca={reul}] )
@ENGLISH:
    look [class=verb] (
      attr at [class=preposition] (
        ii $X ))
@-2xLOG_LIKELIHOOD: 12.77
```

Figure 2: Transfer rule for English lexicalization and preposition insertion

4 Transfer Rule Induction

The transfer rule induction process has the following steps described below (additional details can also be found in (Lavoie et. al., 2001)):

- Nodes of the corresponding source and target parses are aligned using the baseline transfer dictionary and some heuristics based on the similarity of part-of-speech and syntactic context.
- Transfer rule candidates are generated based on the sub-patterns that contain the corresponding aligned nodes in the source and target parses.
- The transfer rule candidates are ordered based on their likelihood ratios.
- The transfer rule candidates are filtered, one at a time, in the order of the likelihood ratios, by removing those rule candidates that do not produce an overall improvement in the accuracy of the transferred parses.

Figures 2 and 3 show two sample induced rules. The rule formalism uses notation similar to the syntactic dependency notation shown in Figure 1, augmented with variable arguments prefixed with $ characters. These two lexico-structural rules can be used to transfer a Korean syntactic representation for *ci-to-reul po-ra* to an English syntactic representation for *look at the map*. The first rule lexicalizes the English predicate and inserts the corresponding preposition while the second rule inserts the English imperative attribute.

4.1 Aligning the Parse Nodes

To align the nodes in the source and target parse trees, we devised a new dynamic programming

```
@KOREAN:
    $X [class=vbma ente={ra}]
@ENGLISH:
    $X [class=verb mood=imp]
@-2xLOG_LIKELIHOOD: 33.37
```

Figure 3: Transfer rule for imperative forms

alignment algorithm that performs a top-down, bidirectional beam search for the least cost mapping between these nodes. The algorithm is parameterized by the costs of (1) aligning two nodes whose lexemes are not found in the baseline transfer dictionary; (2) aligning two nodes with differing parts of speech; (3) deleting or inserting a node in the source or target tree; and (4) aligning two nodes whose relative locations differ.

To determine an appropriate part of speech cost measure, we first extracted a small set of parse pairs that could be reliably aligned using lexical matching alone, and then based the cost measure on the co-occurrence counts of the observed parts of speech pairings. The remaining costs were set by hand.

As a result of the alignment process, alignment id attributes (`aid`) are added to the nodes of the parse pairs. Some nodes may be in alignment with no other node, such as English prepositions not found in the Korean DSyntS.

4.2 Generating Rule Candidates

Candidate transfer rules are generated by extracting source and target tree sub-patterns from the aligned parse pairs using the two set of constraints described below.

4.2.1 Alignment constraints

Figure 4 shows an example alignment constraint. This constraint, which matches the structural patterns of the transfer rule illustrated in Figure 2, uses the `aid` alignment attribute to indicate that in a Korean and English parse pair, any source and target sub-trees matching this alignment constraint (where $X1 and $Y1 are aligned, i.e. have the same `aid` attribute values, and where $X2 and $Y3 are aligned) can be used as a point of departure for generating transfer rule candidates. We suggest that alignment constraints such as this one can be used to define most of the possible syntactic divergences between languages (Dorr, 1994), and that only a handful of them are necessary for two given languages (we have identified 11 general alignment constraints necessary for Korean to English transfer so far).

```
@KOREAN:
    $X1 [aid=$1] (
      $R1 $X2 [aid=$2] )
@ENGLISH:
    $Y1 [aid=$1] (
      $R2 $Y2 (
        $R3 $Y3 [aid=$2] ) )
```

Figure 4: Alignment constraint

Each node of a candidate transfer rule must have its relation attribute (relationship with its governor) specified if it is an internal node, otherwise this relation must not be specified:

e.g.

```
    ⊘ $X1 ( $R $X2 )
```

Figure 5: Independent attribute constraint

4.2.2 Attribute constraints

Attribute constraints are used to limit the space of possible transfer rule candidates that can be generated from the sub-trees satisfying the alignment constraints. Candidate transfer rules must satisfy all of the attribute constraints. Attribute constraints can be divided into two types:

- *independent attribute constraints*, whose scope covers only one part of a candidate transfer rule and which are the same for the source and target parts;
- *concurrent attribute constraints*, whose scope extends to both the source and target parts of a candidate transfer rule.

Figures 5 and 6 give examples of an independent attribute constraint and of a concurrent attribute constraint. As with the alignment constraints, we suggest that a relatively small number of attribute constraints is necessary to generate most of the desired rules for a given language pair.

4.3 Ordering Rule Candidates

In the next step, transfer rule candidates are ordered as follows. First, they are sorted by their decreasing log likelihood ratios. Second, if two or more candidate transfer rules have the same log likelihood ratio, ties are broken by a specificity heuristic, with the result that more general rules are ordered ahead

In a candidate transfer rule, inclusion of the lexemes of two aligned nodes must be done concurrently:

```
e.g.
        $X [aid=$1]
    and
        $Y [aid=$1]
e.g.
        ∅ [aid=$1]
    and
        ∅ [aid=$1]
```

Figure 6: Concurrent attribute constraint

of more specific ones. The specificity of a rule is defined to be the following sum: the number of attributes found in the source and target patterns, plus 1 for each for each lexeme attribute and for each dependency relationship. In our initial experiments, this simple heuristic has been satisfactory.

4.4 Filtering Rule Candidates

Once the candidate transfer rules have been ordered, error-driven filtering is used to select those that yield improvements over the baseline transfer dictionary. The algorithm works as follows. First, in the initialization step, the set of accepted transfer rules is set to just those appearing in the baseline transfer dictionary. The current error rate is also established, by applying these transfer rules to all the source structures and calculating the overall difference between the resulting transferred structures and the target parses, using a tree accuracy recall and precision measure (determined by comparing the features and dependency relationships in the transferred parses and corresponding target parses). Then, in a single pass through the ordered list of candidates, each transfer rule candidate is tested to see if it reduces the error rate. During each iteration, the candidate transfer rule is provisionally added to the current set of accepted rules and the updated set is applied to all the source structures. If the overall difference between the transferred structures and the target parses is lower than the current error rate, then the candidate is accepted and the current error rate is updated; otherwise, the candidate is rejected and removed from the current set.

4.5 Discussion of Induced Rules

In our experiments, the alignment constraints yielded 22,881 source and target sub-tree pairs from the training set of 1,433 parse pairs. Using the attribute constraints, an initial list of 801,674 transfer rule candidates was then generated from these sub-tree pairs. The initial list was subsequently reduced to 32,877 unique transfer rule candidates by removing duplicates and by eliminating candidates that had the same source pattern as another candidate with a better log likelihood ratio. After filtering, 2,133 of these transfer rule candidates were accepted. We expect that the number of accepted rules per parse pair will decrease with larger training sets, though this remains to be verified.

```
@KOREAN:
    {po} [class=vbma ente={ra}] (
      s1 $X [ppca={reul}] )
@ENGLISH:
    look [class=verb mood=imp] (
      attr at [class=preposition] (
        ii $X ) )
@-2xLOG_LIKELIHOOD: 11.40
```

Figure 7: Transfer rule for English imperative with lexicalization and preposition insertion

The rule illustrated in Figure 3 was accepted as the 65th best transfer rule with a log likelihood ratio of 33.37, and the rule illustrated in Figure 2 was accepted as the 189th best transfer rule candidate with a log likelihood ratio of 12.77. An example of a candidate transfer rule that was not accepted is the one that combines the features of the two rules mentioned above, illustrated in Figure 7. This transfer rule candidate had a lower log likelihood ratio of 11.40; consequently, it is only considered after the two rules mentioned above, and since it provides no further improvement upon these two rules, it is filtered out.

In an informal inspection of the top 100 accepted transfer rules, we found that most of them appear to be fairly general rules that would normally be found in a general syntactic-based transfer dictionary. In looking at the remaining rules, we found that the rules tended to become increasingly corpus-specific.

The induction results were obtained using a Java implementation of the induction component. Microsoft SQL Server was used to count and deduplicate the rule candidates. The data preparation and induction processes took about 12 hours on a 300 MHz PC with 256 MB RAM.

5 Evaluation

5.1 Systems Compared

Babelfish As a first baseline, we used Babelfish from Systran, a commercial large coverage MT system supporting Korean to English translation. This system was not trained on our corpus.

GIZA++/RW As a second baseline, we used an off-the-shelf statistical MT system, consisting of the ISI ReWrite Decoder (Germann et al., 2001) together with a translation model produced by GIZA++ (Och and Ney, 2000) and a language model produced by the CMU Statistical Language Modeling Toolkit (Clarkson and Rosenfeld, 1997). This system was trained on our corpus only.

Lex Only As a third baseline, we used our system with the baseline transfer dictionary as the sole transfer resource.

Lex+Induced We compared the three baseline systems against our complete system, using the baseline transfer dictionary augmented with the induced transfer rules.

We ran each of the four systems on the test set of 50 Korean sentences described in Section 3.2 and compared the resulting translations using the automatic evaluation and the human evaluation described below.

5.2 Automatic Evaluation Results

For the automatic evaluation, we used the Bleu metric from IBM (Papineni et al., 2001). The Bleu metric combines several modified N-gram precision measures (N = 1 to 4), and uses brevity penalties to penalize translations that are shorter than the reference sentences.

Table 5 shows the Bleu N-gram precision scores for each of the four systems. Our system (Lex+Induced) had better precision scores than each of the baseline systems, except in the case of 4-grams, where it slightly trailed Babelfish. The statistical baseline system (GIZA++/RW) performed poorly, as might have been expected given the small amount of training data.

Table 6 shows the Bleu overall precision scores. Our system (Lex+Induced) improved substantially over both the Lex Only and Babelfish baseline systems. The score for the statistical baseline system (GIZA++/RW) is not meaningful, due to the absence of 3-gram and 4-gram matches.

System	1-g Prec	2-g Prec	3-g Prec	4-g Prec
Babelfish	0.3814	0.1207	0.0467	0.0193
GIZA++/RW	0.1894	0.0173	0.0	0.0
Lex Only	0.4234	0.1252	0.0450	0.0145
Lex+Induced	0.4725	0.1618	0.0577	0.0185

Table 5: Bleu N-gram precision scores

System	Bleu Score
Babelfish	0.0802
GIZA++/RW	NA
Lex Only	0.0767
Lex+Induced	0.0950

Table 6: Bleu overall precision scores

5.3 Human Evaluation Results

For the human evaluation, we asked two English native speakers to rank the quality of the translation results produced by the Babelfish, Lex Only and Lex+Induced, with preference given to fidelity over fluency. (The translation results of the statistical system were not yet available when the evaluation was performed.) A rank of 1 was assigned to the best translation, a rank of two to the second best and a rank of 3 to the third, with ties allowed.

Table 7 shows the pairwise comparisons of the three systems. The top section indicates that the Babelfish and Lex Only baseline systems are essentially tied, with neither system preferred more frequently than the other. In contrast, the middle and bottom sections show that our system improves substantially over both baseline systems; most strikingly, our system (Lex+Induced) was preferred almost 20% more frequently than the Babelfish baseline (46% to 27%, with ties 27% of the time).

System Pair Comparison	Result
Babelfish *better than* Lex Only	37%
Lex Only *better than* Babelfish	36%
Babelfish *same as* Lex Only	27%
Babelfish *better than* Lex+Induced	27%
Lex+Induced *better than* Babelfish	46%
Babelfish *same as* Lex+Induced	27%
Lex Only *better than* Lex+Induced	18%
Lex+Induced *better than* Lex Only	41%
Lex Only *same as* Lex+Induced	41%

Table 7: Human evaluation results

6 Conclusion and Future Work

In this paper we have described the design of an MT system based on lexico-structural transfer rules induced from parsed bitexts. In a small scale experiment with Korean to English translation, we have demonstrated a substantial improvement over three baseline systems, including a nearly 20% improvement in the preference rate for our system over Babelfish (which was not trained on our corpus). Although our experimentation was aimed at Korean to English translation, we believe that our approach can be readily applied to other language pairs.

It remains for future work to explore how well the approach would fare with a much larger training corpus. One foreseeable problem concerns the treatment of lengthy training sentences: since the number of transfer rule candidates generated grows exponentially with the size of the parse tree pairs, refinements will be necessary in order to make use of complex sentences; one option might be to automatically chunk longer sentences into smaller units.

Acknowledgements

We thank Richard Kittredge for helpful discussion, Daryl McCullough and Ted Caldwell for their help with evaluation and Fei Xia for her assistance with the automatic conversion of the phrase structure parses to syntactic dependency representations. We also thank Chung-hye Han, Chulwoo Park, Martha Palmer, and Joseph Rosenzweig for the handcrafted Korean-English transfer dictionary, and Graham Russell for the corpus-based extracted transfer dictionary. This work has been partially supported by DARPA TIDES contract no. N66001-00-C-8009.

References

Philip Clarkson and Ronald Rosenfeld. 1997. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proceedings of Eurospeech'97*.

Bonnie Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–635.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu and Kenji Yamada. 2001. Fast Decoding and Optimal Decoding for Machine Translation. In *Proceedings of ACL'01*, Toulouse, France.

Chung hye Han, Benoit Lavoie, Martha Palmer, Owen Rambow, Richard Kittredge, Tanya Korelsky, Nari Kim, and Myunghee Kim. 2000. Handling Structural Divergences and Recovering Dropped Arguments in a Korean-English Machine Translation System. In *Proceedings of the Fourth Conference on Machine Translation in the Americas (AMTA'00)*, Mision Del Sol, Mexico.

Daniel Karp and Yves Schabes. 1992. A Wide Coverage Public Domain Morphological Analyzer for English. In *Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING'92)*, pages 950–955, Nantes, France.

Benoit Lavoie and Owen Rambow. 1997. RealPro – A Fast, Portable Sentence Realizer. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'97)*, Washington, DC.

Benoit Lavoie, Richard Kittredge, Tanya Korelsky, and Owen Rambow. 2000. A framework for MT and multilingual NLG systems based on uniform lexico-structural processing. In *Proceedings of ANLP/NAACL 2000*, Seattle, Washington.

Benoit Lavoie, Michael White, and Tanya Korelsky. 2001. Inducing Lexico-Structural Transfer Rules from Parsed Bitexts. In *Proceedings of the ACL 2001 Workshop on Data-driven Machine Translation*, pages 17–24, Toulouse, France.

Christopher D. Manning and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

H. Maruyama and H. Watanabe. 1992. Tree cover Search Algorithm for Example-Based Translation. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'92)*, pages 173–184.

Yuji Matsumoto, Hiroyuki Hishimoto, and Takehito Utsuro. 1993. Structural Matching of Parallel Texts. In *Proceedings of the 31st Annual Meetings of the Association for Computational Linguistics (ACL'93)*, pages 23–30.

Igor Mel'čuk. 1988. *Dependency Syntax*. State University of New York Press, Albany, NY.

Adam Meyers, Roman Yangarber, Ralph Grishman, Catherine Macleod, and Antonio Moreno-Sandoval. 1998. Deriving Transfer Rules from Dominance-Preserving Alignments. In *Proceedings of COLING-ACL'98*, pages 843–847.

Makoto Nagao. 1984. A framework of a mechanical translation between Japenese and English by analogy principle. In A. Elithorn and R. Banerji, editors, *Artificial and Human Intelligence*. NATO Publications.

Alexis Nasr, Owen Rambow, Martha Palmer, and Joseph Rosenzweig. 1997. Enriching lexical transfer with cross-linguistic semantic features. In *Proceedings of the Interlingua Workshop at the MT Summit*, San Diego, California.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL'00*, pages 440–447, Hongkong, China.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. In *IBM technical report, RC22176*.

Stephen D. Richardson, William B. Dolan, Arul Menezes, and Monica Corston-Oliver. 2001. Overcoming the Customization Bottleneck using Example-based MT. In *Proceedings of the ACL 2001 Workshop on Data-driven Machine Translation*, pages 9–16, Toulouse, France.

Satoshi Sato and Makoto Nagao. 1990. Toward Memory-based Translation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING'90)*, pages 247–252.

Fei Xia and Martha Palmer. 2001. Converting Dependency Structures to Phrase Structures. In *Notes of the First Human Language Technology Conference*, San Diego, California.

Juntae Yoon, Seonho Kim, and Mansuk Song. 1997. New parsing method using global association table. In *Proceedings of the 5th International Workshop on Parsing Technology*.