

Cross linguistic Annotator for Icelandic, Hindi, and Spanish

1 INTRODUCTION

Our system entailed inducing a cross-linguistic Part-Of-Speech tagger from Interlinear Glossed Text (IGT) projections and RAW (unannotated) lines of text in a given language. After extracting morphological and distributional clues from the two training sets (IGT and RAW), we used a Hidden Markov Model with Viterbi smoothing and post-processing heuristics to assign the most probable tags to a batch of lines (test data). The tagged words were then evaluated against the gold standard to produce an accuracy figure.

This write-up deals with implementing the final model of our system on Icelandic, the challenge language. During this course (LING 573), we were given the task of constructing a generalized model for tagging two languages of our choice. We picked Hindi and Spanish as our high density and low density languages, respectively. During the last two weeks, we were given the challenge language in order to evaluate our tagger's performance on a third language. We have been successful in tagging all three languages with varying degrees of accuracy. Our baseline involved tagging all words "NN," the most frequent tag in all three languages. Using our tagger we achieved an **improvement** of 16% in Hindi, 12% in Spanish and 39% in Icelandic.

In a nutshell, the system component described in this paper is a "black box", implying minimal constraints on the design specifications and immense number of implementation choices. This translates to a considerable amount of trial and error, discarding ideas, and deploying new ones in order to create a tagger that performs better than the baseline, as described below.

2 OVERVIEW

The steps in our final system are described in pseudocode below, with greater detail in the sections to follow.

- Gather IGT and raw text in target language.
- Clean IGT, automatically and manually.
- Project part of speech tags from translation line of IGT to source line of IGT.
- Induce lexicon over raw text.
 - o Create hash tables over IGT tagged source line and corresponding POS tag for:
 - whole word
 - last 4 characters of the word
 - last 3 characters of the word
 - last 2 characters of the word
 - o For each incoming word from the raw text, check word in IGT lexicon to see if it matches:

- punctuation or numbers, force tag accordingly
- the whole word, assign highest frequency tag from IGT for this word
- else, the last 4 characters of the word, assign highest frequency tag from IGT for this word
- else, the last 3 characters of the word, assign highest frequency tag from IGT for this word
- else, the last 2 characters of the word, assign highest frequency tag from IGT for this word
- All remaining words are tagged NN, for the greatest chance of being the correct default tag
- Result is rule-based tagged version of the raw corpus, to be input to the HMM
- Create HMM over tagged raw corpus by computing transition and emission probabilities
- Format test file to contain end-of-line tag (EOL)
- Run Viterbi tagger over test file, using emission and transition probabilities created in HMM
- Post-process—attempt to “clean up” mistakes made by Viterbi
 - Force tag punctuation and numbers
 - Check all tags against the tags assigned in the induced lexicon, assuming Viterbi will handle unknowns, and the raw text induced lexicon will act as a “gold standard” for knowns.

We evaluated our system against Hindi, Spanish and Icelandic. We do not implement any language-specific processing whatsoever. As mentioned in the results section (section 4), we have determined that the quality of IGT in a given language greatly determines our tagger’s performance and is responsible for the varying degrees of accuracy.

3 SPECIFICATIONS

We describe below the steps undertaken in our POS Annotator. A few experiments were conducted on variations of the basic implementation design, the results of which are included in the Experiments section.

3.1 DATA FILES

The starting point for our project is availability of two input files, namely IGT (Table 1) and RAW (Table 2), for a given target language.

The IGT file contains a source line, gloss line, and translation line. We tagged the English translation lines using the Ratnaparkhi Tagger. We projected these tags onto the language lines via the gloss words to obtain tagged source lines (one sentence per line) extracted from IGT sequences in the following format:

\$line_num \$word \$tag \$word \$tag \$word \$tag \$word \$tag
\$line_num \$word \$tag \$word \$tag \$word \$tag ...

Example from Hindi:

1681 Raam-ne_NN chIIkaa_VBD

Example from Spanish:

1210 Escrib-e_VBG una_DT novela_NN

Example from Icelandic:

2224 Batinn_NN rak_VBD a_TO land_NN

| | HINDI | SPANISH | ICELANDIC |
|-----------------|-------|---------|-----------|
| # IGT instances | 562 | 1,092 | 754 |

Table 1: Number of IGT instances (sentences) used in experiments

The RAW file contains unannotated lines of text with one constraint: each sentence must end with a period (an exclamation mark ‘!’ or a question mark ‘?’ will also work), and there must be at least one blank space between each word, including any sort of punctuation marks. We have a PERL script to take care of this sort of preprocessing. The format is as follows:

Example from Hindi:

isa xurgAWApA patta ke kenxra meM aRtaBujAXArI - xurgA (CawravAlli xevI) Ora usake vAhana xo siMha ciwriwa kiye jAwe hEM .

Example from Spanish:

PEKIN , Ene 2 (AFP) - El año 1995 terminó en China con 48 condenas a muerte señaladas por los diarios del martes en Pekín , lo que llevó el número de penas capitales pronunciadas en las últimas semanas a aproximadamente 150 .

Example from Icelandic:

Ef til vill er þetta bara orsök þess að vera heima og læra og hitta varla annað fólk nema sammendur þar sme maður talar lítið um annað en námið , og er með klúr komment hægri vinstri .

| | HINDI | SPANISH | ICELANDIC |
|------------------|--------------|--------------|--------------|
| Number of lines | 160 thousand | 292 thousand | 308 thousand |
| Number of tokens | 3.7 million | 3.2 million | 773 thousand |
| Number of types | 115 thousand | 97 thousand | 70 thousand |

Table 2: Approximate size of RAW text used in experiments

We also have a gold standard file (Table 3) for each of our languages. The GOLD file contains the words and their tags extracted from an annotated corpus, one entry per line.

For Icelandic, we were provided with a development corpus for testing purposes. The actual gold corpus was not given to us at the time of writing this paper; before the challenge language contest. Thus the figures for Icelandic in Table 3 refer to the corpus we used for experiments (10% of the development corpus). The format of the GOLD file is as follows:

\$word *\$tag*
\$word *\$tag*

Example from Hindi:

megamana *NN*
 [*PUNC*

| | HINDI | SPANISH | ICELANDIC |
|-------------------------|--------------|-------------|-------------|
| Number of tokens | 522 thousand | 10 thousand | 84 thousand |
| Number of types | 12 thousand | 2 thousand | 14 thousand |

Table 3: Approximate number of words to be tagged in testing phase

The GOLD file is used for evaluating the results of our tagger. These results are obtained by running our tagger on testing data (TEST, Table 4). The content of the test file is same as the gold standard file minus the tags.

| | HINDI | SPANISH | ICELANDIC |
|----------------------------------|-----------------------|------------------------|---------------------|
| Source | ANNCORRA Hindi Corpus | CRATER En-Sp-Fr Corpus | IS_DEVE LOPMENT.txt |
| Size relative to training | 10 percent | 0.3 percent | 10 percent |

Table 4: Details of the testing data used in our experiments

The TEST file for Hindi was extracted from the same corpus as the training set. The ANNCORRA Hindi Corpus had Hindi tags often numbering to more than one for each word. We computed a mapping table to our 13 tags by first translating the Hindi POS tags into English. As 90% of the data was used for training, the testing set was set at around 10 percent.

The Spanish test set was extracted from the Crater corpus, a small bitext corpus found online with approximately 9000 annotated words. The tags used were not PENN Treebank tags, but we were able to infer a relative mapping to our 13 tags based on comparing it with the parallel text. Even though the test set was not from the same corpus as the training set, we did still have decent results for Spanish. Note the reason for such a small amount of testing data for Spanish is that by definition, Spanish was our low-density language in this project and hence did not have a tagged corpus per se. We were able to extract a fraction from the net. We did not use this resource in any way in the training phase, as per the rules of the project.

In the Icelandic test corpus, we were already given the tags mapped to our predetermined universal set of 13 tags (Table 5).

| TAG | DESCRIPTION |
|------|---------------------------|
| PUNC | punctuation |
| NN | nouns |
| SCC | short closed class |
| JJ | adjectives |
| RB | adverbs |
| VB | verbs |
| IN | adposition |
| CD | cardinal number |
| CC | coordinating conjunctions |
| IND | indexicals |
| DT | determiners |
| QW | question words |
| MISC | miscellaneous |

Table 5: Set of tags used by our tagger

3.2 IGT PROJECTION

This phase of the project included collecting Interlinear Glossed Text for our target language, cleaning it of noise, and projecting tags on the source line to be used in the next phase. Our primary source of IGTs was ODIN. For Hindi and Spanish, we also crawled the web to find more IGT instances in linguistic research papers. For more details, see Table 1.

Under “Projecting Annotations,” we started with usable IGTs in the specific language. We define “usable” IGTs as a sequence of interlinear glossed texts that are 3 lines each (corresponding to source, gloss, and translation lines). These lines have also been screened to discard non-Hindi, non-Spanish, and non-Icelandic text. We then tag (using the Ratnaparkhi Tagger) and stem (using `eng_morph.pl`, provided to us) the translation line, stem the gloss line, project the tags from the translation (3rd line) onto the gloss (2nd line), and finally project the morpho-syntactic tags from the gloss line onto the source line.

3.3 SEED LEXICON

A PERL script reads in all tokens of `$word_$tag` from IGT text and maps all tags to our set of 14 tags. (This set is the pre-determined set of 13 plus EOS which is used to tag periods, question and exclamation marks denoting end of sentence. We remap EOS to PUNC before evaluation against the gold standard). If a sentence does not contain a terminal period, we add one. Next, we record frequencies of tags, words, word with tag, morphemes, and morpheme with tag. Extracting morphemes for the morphological dictionary is a 2-step process. We first parse a word to check for presence of separator

characters like period, colon, equal-to, parenthesis, etcetera, which denotes the boundary between a word and its morpheme. We then record the morpheme and its associated tag. If a morpheme is not found in such a way, we resort to storing the last 4, 3, and 2 characters of the word as a morpheme cue. Also note that there is bound to be noise in our tagged source lines. Hence, we filter out those tags (and thus words) which are not in a predetermined format, for example all uppercase. Some of these details are given in Table 6. (**“Words included” gives the number of words that were actually added from IGT to seed lexicon**).

Example from Hindi:

khariid-ne-ko_VB => The morphemes ‘ne’ and ‘ko’ are associated with the tag VB

Example from Spanish:

estudiante_NN => The last four, three, and two characters “ante,” “nte,” and “te” are associated with the tag NN

Example from Icelandic:

falla_fall-INF => The Icelandic word ‘falla’ is not added in the seed lexicon, because the projected tag is not a valid tag. This example demonstrates some of the noise inherent in IGT projections.

| | HINDI | SPANISH | ICELANDIC |
|-----------------------|-------|---------|-----------|
| Total words | 3,390 | 7,538 | 5,026 |
| words included | 2,265 | 4,827 | 4,103 |

Table 6: Statistics of the Seed Lexicon induced from IGT

3.4 INDUCING LEXICON FROM RAW TEXT

After obtaining a seed lexicon from IGT, we now tag words in the raw text using distributional and morphological cues (Table 7). While we do not have any orthographic heuristics (all words are converted into lower case), we do **force-tag** punctuation marks with PUNC and all numbers with CD. To help us determine the start state probabilities in our Hidden Markov Model, we force-tag periods, question, and exclamation marks with a fourteenth tag EOS (end of sentence). We next check for the presence of a **word in the seed lexicon**. If this does not yield a tag, we cross-check **presence of a morpheme** from the morphological dictionary and record its associated tags. The morpheme checker takes into account the length of a word. Thus we first check for the presence of morphemes of length > 3 followed by length 3 and if that does not suffice, morphemes of length 2. Our reasoning is that longer length morphemes will be better predictors than shorter length ones. If none of the above checks results in a tag, we assign the word (unknown word) the most probable tag in the corpus, currently **hard-coded to NN**. This suits all the three languages.

At this stage, we may have more than one tag assigned to a particular word. In order to ensure each word has at most one tag, we refer to the distributional clues obtained from

the IGT corpus. We determine which tag occurs with the word the most number of times in our IGT instances. Each word is thus assigned the **most frequent tag**.

The first version of our model did not filter tags. We had entries with words assigned multiple tags. Our second approach was to filter tags according to the number of tags a word has. For example if a word has between 3 and 12 tags, we chose the 5 most probable. However this did not improve the performance of our system. Our system continued to perform below the baseline because there was too much ambiguity in the model. Hence we ultimately modified our lexicon such that **each word is now assigned at most 1 tag**.

One important feature of our model is that while it is true that we induce tags on the raw text by using a seed lexicon derived from the IGT text, our HMM is **exclusively modeled on the raw text lexicon**, (i.e. no IGT). This helped improve the performance since the test data resembles the training set more than text in IGT.

| | HINDI | SPANISH | ICELANDIC |
|----------------------------|--------------|--------------|--------------|
| total read (tokens) | 3.7 million | 3.2 million | 773 thousand |
| force tagged | 430 thousand | 495 thousand | 103 thousand |
| in the IGT lexicon | 3.2 million | 2.6 million | 290 thousand |
| matched suffixes | 41 thousand | 45 thousand | 116 thousand |
| unknown (tagged NN) | 55 thousand | 29 thousand | 264 thousand |

Table 7: Details of the lexicon induced over RAW text; Number of words derived from each source

3.5 HIDDEN MARKOV MODEL

We built our HMM over the raw corpus, which had already been tagged using the set of rules described above. The transition probabilities were deduced from the ordering of words in the raw corpus. This was merely a best approximation, because the tags that were input into the HMM were the best guesses that could be deduced from the rules. The transition and emission probabilities for NN were a bit artificially high, given the large number of default NN's in our tagging of the raw corpus. This seemed to be the best of available alternatives, however, as NN is the most frequent tag in the corpus.

This was an improvement over our initial system design, because we built our HMM only over what was seen in the tagged raw corpus—i.e., each word was assigned at most one tag. Our initial model attempted to give all possible tags seen in the morphological analysis phase to all words seen in the raw corpus. This meant that most words had approximately 3-8 of the 13 tags, and all unknowns had equi-probability over all 13 tags. In addition, our tag-tag transitions were initially induced from our IGT corpus, which proved to be too sparse to have much influence in the HMM. Though we tried very hard to hone this previous model, there was too much uncertainty in the HMM to produce any

results. We chose to default to NN for unknowns so that we could give the HMM an entirely tagged corpus.

3.6 VITERBI SMOOTHING

Our preliminary tests with all of our various HMM models were run through the testvit application. We were not able to get it to handle unknown words in any successful capacity. We ran small tests with just one unknown word, and each time, the output sequence disintegrated from a nice variety of numbers to a repeating string of 1's.

We decided to build our own Viterbi code, and found this much more successful and malleable. In particular, we found it helpful to be able to experiment with smoothing and how the system handled unknown words. The input to the code was the entire test file, but the Viterbi process was run on each sentence.

Further post processing on the Viterbi output is described in the next section.

3.7 POST PROCESSING HEURISTICS

After obtaining the results estimated by Viterbi (tags assigned to words from test data), we run a post-processor to reassert certain rules in our tagger. We override the most likely tag assigned to a word by Viterbi in 3 cases:

1. If the word is a punctuation mark, we force tag it PUNC.
2. If the words is a number, we force tag it CD.
3. If the word is present in the lexicon induced over raw text (section 3.4), we force tag it the given tag therein.

We deemed post processing to be essential after we saw that Viterbi has a tendency of “getting stuck” in places, i.e. assigning a sequence of the default tag (for unknown words) even if subsequent strings of words are present in the lexicon. The experiments in section 4 show that post processing increases the accuracy rate by 1-3 percent in all the 3 languages. We also think that since this force tagging is universal—that is, a punctuation and a number will always be assigned PUNC and CD respectively in any language—it justifies this design choice.

4 RESULTS AND EXPERIMENTS

The following tables display the results of final system on the three languages. Each of the below described systems were implemented on the RAW text (section 3.4) and Table 2 and passed into `makeHmm.pl` and `viterbi_tagger.pl` in order to assign a tag sequence to TEST data, which was in turn evaluated against GOLD data.

Table 8 shows our baseline tagger for Hindi, Spanish, and Icelandic. In this system, each and every word in the raw text was assigned the most probable tag (hard coded to noun, NN).

| | HINDI | SPANISH | ICELANDIC |
|-----------|---------|---------|-----------|
| NN Tagger | 34.80 % | 30.26 % | 18.13 % |

Table 8: Baseline -- NN

Table 9 shows the results of our final tagger. The first row evaluates the Viterbi assigned tags against the Gold corpus. The second row evaluates after we perform the force-tagging and other post-processing heuristics (section 3.7). Compare with Table 8 (our baseline). One reason for such a huge discrepancy in the baseline performance of NN-tagging between Icelandic and the other 2 languages is the degree of uncertainty in the tag mapping. All our test corpora were morpho-syntactically richer compared to our predetermined set of tags (Table 5). Therefore we had to create a many-to-one mapping of tags from the test corpora to our set. While the Icelandic set was given to us with the mapping already established, we were in charge of mapping Hindi and Spanish tagset. We feel that we might have been overly generous with mapping to NN, as we set NN as our default tag such that if none of our mapping rules yielded a match, we would translate the tag to NN.

| | HINDI | SPANISH | ICELANDIC |
|---------|---------|---------|-----------|
| VITERBI | 48.75 % | 40.62 % | 57.57 % |
| POST | 50.79 % | 42.99 % | 57.88 % |

Table 9: Final System in Hindi, Spanish, and Icelandic

We reflected on the reason behind such a marked improvement in Icelandic corpus from baseline to final figures which is absent in Hindi and Spanish. Our system is designed such that we perform no language-specific computations. Thus the only difference is in the input data (IGT and RAW text). We have come to the conclusion that the degree of accuracy in IGT source tagged lines is extremely significant. When cleaning the IGT for Icelandic, we were more conscientious than in Hindi and Spanish. This is because we were already familiar with the projection algorithm and thus incorporated certain manual and automated cleaning mechanisms to achieve optimum projection of the tags. We ensured maximum number of word matches between translation and gloss lines, and maximum number of one-to-one correspondences between gloss and source lines.

We performed a few experiments on different flavors of our system. The Hindi and Spanish experiments were performed, evaluated and included in the previous report. Below, we present a few figures from experiments on Icelandic.

We have reached our final system through various trials and errors. We initially started with having no morpheme learner in our raw text induction phase. We learnt from words present in the IGT corpus, and tagged the rest of the words (unknown) as NN. In order to boost this performance we established our post processing heuristics, described in section 3.7. However after we established the suffix learner in our system, post processing no longer boosted the scores by 5-7%. Nevertheless, we retained our heuristics, because it did not harm our performance. In fact it improved our accuracy by a fraction. To show the impact of post processing, we show in Table 10, implementing post processing heuristics on the baseline corpus, wherein all words are tagged NN.

| | ICELANDIC |
|---------|-----------|
| VITERBI | 20.86 % |
| POST | 31.81 % |

Table 10: Impact of Post processing on Base (all NN) tagged corpus

In our final system for Icelandic (RAW – extra), we have added 200 KB of blog-text to increase the volume of raw text to be induced. RAW – base refers to the files given to us on Pongo. However we could discern no significant change in the tagger performance. Note the testing was done on our evaluation corpus (EVAL) and not one our testing corpus (TEST). Details follow in Table 12.

| ICELANDIC | RAW - base | RAW - extra |
|-----------|------------|-------------|
| VITERBI | 56.19 % | 56.26 % |
| POST | 56.52 % | 56.57 % |

Table 11: Impact of Raw Text size on Icelandic Corpus

For Icelandic, we were given IS_development.txt. We included 80% of this file in our raw corpus and divided the remaining 20% into evaluation and test (TEST – 1) corpus. All our experiments were performed on evaluation corpus. After we established the final system, we tested on the test corpus. Finally, we also tested on another corpus, IS_test.txt given to us for the challenge language contest. Note we do not have the gold corpus for this third set. We have included our tagging of the data as an entry to the contest. Details of these corpora follow in Table 12. Our final figure of 57.88% is on the TEST – 1 corpus.

| ICELANDIC | EVALUATION | TEST - 1 | TEST - CONTEST |
|------------------|-------------|-------------|----------------|
| Number of tokens | 84 thousand | 84 thousand | 116 thousand |
| Number of types | 16 thousand | 14 thousand | 15 thousand |

Table 12: Approximate volume of the different Icelandic test corpus

5 DISCUSSION AND POSSIBILITY FOR FUTURE WORK

WHAT WE TRIED THAT DID NOT WORK

- Assigning more than one tag to the induced raw corpus lexicon.

It seemed that allowing the induced lexicon to contain more than one high-frequency tag would enrich the model. What resulted, in our case, was a lexicon where most of the words contained many or most of the tags. This created too much uncertainty in the model, and we saw improvement when we chose only one tag for each word—the highest frequency tag from the IGT lexicon, or NN. However, if a solid method of incorporating multiple possible tags into the lexicon could be devised, we still believe that Viterbi would benefit from this information, and use transition probabilities to sort out the uncertainty between tags.

- Using the transition probabilities only from the IGT.

Our initial model used the transition probabilities only from IGT and the emission probabilities from IGT and the induced lexicon. This proved to be too sparse, and building our HMM over the entire rule-tagged raw lexicon provided our first reasonable results above the baseline. It was not enough simply to induce a lexicon, but we had to preserve the transition probabilities of the raw corpus.

- Assigning the most probable tag over the IGT and induced raw lexicon combined.

At first we made counts over all of the words in the raw lexicon and added them to our initial IGT counts, with counts for all possible tags for each word, before choosing the best tag for the raw words. This proved to skew the numbers artificially, and we found great improvement by choosing the highest frequency tag for each raw word by comparing it to the IGT lexicon alone (both whole word and suffixes).

- Using only whole words and not morphological suffixes.

When we did not compare the raw words to the suffixes of the IGT words, we had a majority of our raw text being tagged as the default NN, and our model was too weak. In addition, it was not enough to check merely the last three characters of a word. We showed great improvement in checking the last four, three, and two characters respectively.

- Making the input to the hmm a series of morphological endings with tags, rather than words with tags.

For example:

walking_VB → ing_VB

This seemed promising, as we thought it would reduce the number of types in the HMM, while still preserving the tag-tag transitions and incorporating morphological information. It didn't perform terribly worse, but it did not improve the model, nor did it provide the breakthrough we were seeking. In addition, we could not incorporate morpheme endings of different lengths as we could when we chose the whole word—instead, we had to choose a specific length, and these are the results:

4 characters: 51.76%

3 characters: 52.72%

2 characters: 51.13%

- Adjusting emission and transition probabilities in Viterbi

In an attempt to smooth the probabilities over unknown words in the test set, we tried to adjust the emission probabilities in Viterbi to so that unknown words were assigned only more open class tags: NN, VB, JJ, and RB. This showed improvement in small test sets, but not on the overall corpus. In addition, we tried having Viterbi automatically reduce the transition probability for NN|NN, as we knew we would have an unnatural amount of nouns coming in from the training corpus (as NN is our default tag). This decreased the overall accuracy by about 2%.

- Edit distance.

We thought that after checking against the IGT lexicon and morphological analyzer, we could try one last step of finding the smallest edit distance between the incoming raw word and the words in the IGT lexicon. We used the Levenshtein Distance Algorithm, for which code is available online in many programming languages. This proved to decrease our final test accuracy slightly to 51.32%. We did not pursue the edit distance further, but it would be interesting to see how it might affect the overall process if used at a different stage, or instead of our lexical rules.

Due to the fact that we made significant changes to the design, but saw little overall change in the results (most produced accuracy between 50-55%), we have concluded that we have exhausted the knowledge that is available from the IGT provided. It seems that in order to create a real breakthrough in accuracy figures, the next logical step would be to collect more IGT and increase the vocabulary available to the system.

FUTURE WORK

- Changes to Viterbi.

We ran our Viterbi code over each sentence, but it would be interesting to try it over larger sequences, such as several sentences, to determine if there is a sequence threshold where Viterbi performs best. We would also like to try more advanced smoothing techniques. Currently we are assigning a very small probability to all 0.0 frequencies, but there are more advanced algorithms to improve smoothing.

- Baum Welch

We were provided with the umdhmm package for implementing the Baum Welch and Viterbi algorithms. We were not able to get either program to produce successful results; thus, we built our own Viterbi code, but did not build our own Baum Welch code. Due to the fact that we started to reach an accuracy plateau in our results, it seems that another major system component is needed. In addition to gathering more IGT for our lexicon, we believe that a successful Baum Welch implementation would provide improvement in our HMM model.

- Several tags in the induced raw lexicon

As described above, we chose to default to one tag per word in the induced raw lexicon. Clearly, however, a word can be assigned multiple tags, both correctly (in different contexts) and incorrectly (due to errors in IGT projection). Thus, it seems that choosing only the highest frequency tag from the IGT lexicon might be limiting the model in such a way that a perfectly plausible tag for a word may never make it into the HMM. Our attempts to work several tags into the model did not prove fruitful, but we believe that a successful implementation of this design component would in fact improve tagging accuracy from Viterbi. It might present a problem in the post-processing step, as the induced lexicon acts as a sort of gold standard with just one tag; but with improved Viterbi output, the post-processing may be increasingly less necessary.

6 CONCLUSION

The final results for Hindi, Spanish, and Icelandic are as follows (Table13). The final system learns tags of words from IGT based on distributional characteristics (most frequent tag) and morphological clues (morpheme suffixes of four, three, and two characters). After inducing a Hidden Markov Model followed by Viterbi smoothing, the tags are run through a post processing heuristics, force-tagging numbers and punctuation marks, followed by retaining the tags of “known” words in the tagged raw corpus.

| | HINDI | SPANISH | ICELANDIC |
|--------------------------|--------------|-------------|--------------|
| IGT (# instances) | 562 | 1092 | 754 |
| TRAINING (tokens) | 3.7 million | 3.2 million | 773 thousand |
| TESTING (tokens) | 522 thousand | 10 thousand | 84 thousand |
| BASELINE | 34.80 % | 30.26 % | 18.13 % |
| FINAL | 50.79 % | 42.99 % | 57.88 % |

Table 13: Cross-linguistic Tagger for Hindi, Spanish, and Icelandic

As discussed in section 4, we know in this sparse data domain, our tagger induction depends primarily on tagged IGT corpus for learning correct transitions and emissions. Thus the degree of accuracy in tag-projected source lines is most significant for optimum tagger performance. This is evident from the higher model improvement for Icelandic in contrast to that for Hindi and Spanish.

