NLP Course Project Report

Automating Conversion of News Headlines into their canonical form

Anubhav Kumar (17085018), EEE

Arpit Jain (17045027), CHE

Arbabosu Mandal (17095015), ECE

Mentor: Samapika Roy

Table of Content

1. Objective

Automate the conversion of news headlines into its canonical form (grammatically correct sentences).

Ex: Raw headlines: KSERC to evaluate power tariff details
Grammatically transformed: KSERC is going to evaluate power tariff details

2. Introduction

Parallel corpus was prepared by collecting news headlines (raw data) and manually converting them into grammatically correct sentences (transformed data) using a given set of rules. Manual conversion included reordering POS, changing tense forms of verbs, introducing forms of verb 'be', etc... The original raw and transformed corpus is present in the "corpus/original" directory.

Using rule-based methods, SMT or a hybrid approach; this process has to be automated. Such formulations can have application in e-reader services, as an accessibility service, etc..

3. Characteristics of News Headlines *Reference*

News headlines are different grammatically then normal sentences. They -

1. Use present simple tense for past events
   The present tense is quick and current, and helps emphasise the action happening, rather than its completion.
   - Parliament confirms new stray dog policy
   - Lion escapes zoo

2. Leave out auxiliary verbs
   With perfect, continuous and passive structures, auxiliary verbs are not necessary. This makes some headlines appear to be in the past tense, when actually the headlines use past participles, or particles, not the past simple.
   - Four stranded in sudden flood (four people have been stranded / were stranded)
   - Temperatures rising as climate changes (temperatures are rising)

3. Use infinitives for future events

Using the infinitive, a future time is not always necessary to demonstrate the future tense in headlines.

- Parliament to decide new policy tomorrow
- President to visit France for further talks

4. Leave out articles (a, an, the)
    - Prime Minister hikes Alps for charity (The Prime Minister hiked the Alps)
    - Man releases rabid dog in park (A man released a rabid dog in a park)

5. Leave out "to be"
    - Residents unhappy about new road (residents are unhappy)
    - Family of murder victim satisfied with court decision (family of murder victim is satisfied.)

6. Leave out "to say"
    - Mr Jones: "They're not taking my house!"

7. Replace conjunctions with punctuation
    - Police arrest serial killer – close case on abductions
    - Fire in bakery; hundreds dead

8. Use figures for numbers
    - 9 dead in glue catastrophe
    - 7 days to Christmas – shoppers go mad

4. Corpus Preparation

Following changes are made in the original corpus present in the "corpus/original" directory -

1. Aligned mismatched raw headlines & their corresponding transformation
2. Some data with only raw, or transformed headlines were pruned
3. Certain nouns and their abbreviations were converted to a common word in the corpus to reduce data sparsity (Chief Minister – CM; Govt – Government)
4. Due to ambiguity in the original corpus, punctuation was removed from sentences to have uniformity

After making these changes, the prepared corpus is thus present in the "corpus/prepared" directory for use in model creation.

5. 1st Attempt - IBM Model 1

Directory -- “1st_attempt_collabotory”
To build model -- Follow instructions in “1st_attempt_collabotory/README.txt”

1. The corpus was divided into 3 subsets, training, dev and test. The model was trained by first dividing the sentences into words and creating 2 vocabs. Then probabilities were calculated by tokenising and using Standard Maximization Algorithm. Convergence was reached in 35 iterations
2. Alignments for the first words of particular English sentences evaluated using the trained translation probabilities of IBM Model 1 following ‘HMM based Word Alignment in Statistical Translation’
3. First order transition probabilities initialized using the technique outlined in ‘Word Alignment for Statistical Machine Translation Using Hidden Markov Models’
4. Instead of capturing the absolute positions for word alignments, only the relative positions i.e the jump widths are taken into consideration
5. Using Bigram Models to create coherent translations
6. Greedily generate the translated sentence based on the above components

5.1 Model Creation

1. Transfer all the files from the folder“1st_attempt_collabotory” in a collab page
2. Then run the “1st_attempt_SMT_headline_to_canonical.ipynb” file

5.2 Conclusion

Bleu score was too low in this approach(0.47) so we decided to try using Moses and also parse the sentences and training the model to find and fix error in the parsed sentence as discussed below

6. Final_attempt (Moses - Model 2 & 3)

Directory -- “Final_attempt(moses)”
To build model -- Follow instructions in “Final_attempt(moses)/README.txt”

Moses is an implementation of the statistical (or data-driven) approach to machine translation (MT). This is the dominant approach in the field at the moment, and is employed by the online translation systems deployed by the likes of Google and Microsoft. In statistical machine translation (SMT), translation systems are trained on large quantities of parallel data (from which the systems learn how to translate small segments), as well as even larger quantities of monolingual data (from which the systems learn what the target language should look like).

The training process in Moses takes in the parallel data and uses co-occurrences of words and segments (known as phrases) to infer translation correspondences between the two languages of interest. Two models supported by moses will be used for the given news headline transformation as follows:

1. **Model 2 (Moses - PBSMT)** :- correspondences are simply between continuous sequences of words

2. **Model 3 (Syntax)** :- more structure is added to the correspondences by introducing linguistic info

6.1 Data Pre-processing

1. Randomly split parallel corpus present in “corpus/prepared/corpus.tsv“ into training, validate & test sets in the ratio 75-15-15 using the python script “Final_attempt(moses)/moses-model23/split.py“
2. Data Files created are train.rw, train.tr, validate.rw, validate.tr, test.rw, test.tr where rw=raw, tr=transformed news headlines
3. Data files are tokenized and truecased using moses scripts [Reference]
4. Prepared corpus files train.clean.rw, train.clean.tr, validate.clean.rw, validate.clean.tr, test.clean.rw, test.clean.tr are present in “Final_attempt(moses)/moses-model23/split“ directory for use in model creation

6.2 Bash Script for Model creation

Follow steps in “Final_attempt(moses)/README.txt“ to create model and get transformed headlines.

It will automatically install all the dependencies, train/tune/test model2 (moses-pbsmt) & model3 (moses-syntax) and generate BLEU scores against test.clean.rw corpus

6.3 Observation

Execution of bash script “run-model23.sh“ will create test.model2.moses-pbsmt.transformed.tr, test.model3.moses-syntax.transformed.tr files in “moses-model23/split“ directory.

Based on results from “Final_attempt(moses)/Output-exampleRun“ --

Key features of Model 2 (Moses - PBSMT) are:

1. BLEU score = 0.72
2. Output of test.clean.tr is produced in ”moses-model23/split/test.model2.moses-pbsmt.transformed.tr”
3. Despite of high BLEU score, the output produced lacks reordering and association like “: ⇔ said”

Key features of Model 3 (Moses - Syntax) are:

1. BLEU score = 0.70
2. Output of test.clean.tr is produced in ”moses-model23/split/test.model3.moses-syntax.transformed.tr”
3. More acceptable transformation as compared to Model 2 due to presence of POS tagged parsed input and output training data
4. Lacks in insertion of absent grammatical features like articles, aux verbs

6.4 Conclusion

1. Model 2 lacks reordering and association due to insufficient corpus size and lack of any linguistic info

2. Introduction of linguistic info in the form of POS tags and parse trees leads to more acceptable transformation by Model 3. It however still lacks in inserting grammatical features like articles, aux verbs due to insufficient size & poor quality of available corpus