**Introduction**

- BBN's statistical translation system for Patent MT
  - Initially developed for newswire, and later for broadcast news, web forums, etc.
  - Best performing system in MT evaluations under DARPA’s GALE, BOLT, and other MT-related programs
  - All techniques initially developed for other domains work well on patents
  - Special handling for patents helps
  - Lots of potential
    - Patents are easier to translate
    - State-of-the-art accuracies in both automatic and manual evaluations
    - Helpful in real patent examination and possibly other tasks

**Statistical Machine Translation**

- **Translation framework**
  - Parallel Text
  - Target Language Text
    - Rule Extraction
    - LM Estimation
    - Chart Parsing
    - K-best Generation
    - Feature Extraction
    - With Extended Features
    - 1-Best Translation
    - Decoding
    - Re-ranking

- **String-to-dependency hierarchical translation model**
  - Extract only hierarchical rules with well-formed dependencies on the target side:
    \[ X_1 : X_r \xrightarrow{\text{POS tag}} X_r \xrightarrow{\text{NB}} X_N \]
    - Use POS tag of head word as non-terminal labels on the target side
    - Extract all phrasal rules, ignoring dependency
  - **Features:**
    - 10+ core features
    - ~50K sparse binary features

**Application to Patent MT**

- **Data preparation**
  - Parallel data: 45M words of Chinese-English sentence pairs
  - Extra LM data: 14B words of US patents in English
  - Development data: 2K Chinese-English sentence pairs, split into tuning and test sets
- **Model training**
  - Translation model: trained on the 45M parallel corpus
  - Language models:
    - 45M LM: trained on the target side of the 45M parallel corpus
    - 14B LM: trained on the 45M LM data plus 14B English patents
  - Addressing issues related to patent data (NTCIR-9):
    - Consistent tokenizer of ASCII strings in source and target
      - E.g., “IS-1000” vs. “IS – 1000”
    - Special token sharing in translation and language model
      - One special token for each category: numbers (e.g., 2,596), e.g., “IS-1000” vs. “IS – 1000”
  - **Patent case-LM**
    - Retrain the case-LM on 45M LM data
    - Word segmentation lexicon
    - Re-optimize on 45M parallel corpus
    - Use only 100 features of the highest weights in each category of the 50K sparse features
    - Address over-fitting due to smaller tuning set
  - **Document-level CLIR-based LM adaptation**
    - Retrieve most relevant passages for a test document in the 14B LM data using CLIR
    - Bias LM for sentences in the test document to these passages

**Recent Advances**

- **Mistakes features**
  - Model target bigrams given source and vise versa
  - Trait features: model general properties of translation hypotheses, e.g., percent of words that re-order
- **Sentence-level LM adaptation instead of document-level**
  - Patent documents tend to use well-structured sentences and re-use n-grams in other patent documents
  - Extra LM data: 14B words of US patents in English
  - Extract all phrasal rules, ignoring dependency
- **Bias LM for sentences in the test document to these passages**
  - Helpful in real patent examination and possibly other tasks

**Results**

- **On development set**
  - Development in NTCIR-9 with 45M LM
    - BLEU: 37.71
    - + translation-based true caser: 41.19
    - + recurrent neural network LM: 42.13
    - + translation-based true caser
    - + recurrent neural network LM
  - NTCIR-9 system with 14B LM
    - BLEU: 39.14
    - + translation-based true caser: 41.09
    - + recurrent neural network LM: 42.13
    - + translation-based true caser

- **Official evaluation**
  - Automatic Evaluation (BLEU)
    - System: BBN-1
      - BLEU: 39.98
    - BBN-2
      - BLEU: 42.68
    - Baseline1
      - BLEU: 36.69
    - Baseline2
      - BLEU: 36.95
      - + translation-based true caser: 41.09
    - + recurrent neural network LM: 42.13
    - + translation-based true caser

- **Typical example**
  - Source: 对于每一像素，变焦引擎210使用在以上等式(2)和(4)中的加权来确定所
    - 该像素是否在三角形内
  - MT output: For each pixel, the rendering engine 210 uses the edge equation set forth in equations (2) to (4) above to determine whether the pixel is in a triangle
  - Reference: For each pixel, the shading engine 210 determines whether the pixel is in the triangle using the edge equations set forth in equations (2) to (4) above.

---

**Statistical Machine Translation**

- **Translation framework**
  - **Translation framework**
    - **Parallel Text**
      - **Target Language Text**
        - **Rule Extraction**
        - **LM Estimation**
        - **Chart Parsing**
        - **K-best Generation**
        - **Feature Extraction**
        - **With Extended Features**
        - **1-Best Translation**
        - **Decoding**
        - **Re-ranking**

- **String-to-dependency hierarchical translation model**
  - Extract only hierarchical rules with well-formed dependencies on the target side:
    \[ X_1 : X_r \xrightarrow{\text{POS tag}} X_r \xrightarrow{\text{NB}} X_N \]
    - Use POS tag of head word as non-terminal labels on the target side
    - Extract all phrasal rules, ignoring dependency
  - **Features:**
    - 10+ core features
    - ~50K sparse binary features

**Application to Patent MT**

- **Data preparation**
  - Parallel data: 45M words of Chinese-English sentence pairs
  - Extra LM data: 14B words of US patents in English
  - Development data: 2K Chinese-English sentence pairs, split into tuning and test sets
- **Model training**
  - Translation model: trained on the 45M parallel corpus
  - Language models:
    - 45M LM: trained on the target side of the 45M parallel corpus
    - 14B LM: trained on the 45M LM data plus 14B English patents
  - Addressing issues related to patent data (NTCIR-9):
    - Consistent tokenizer of ASCII strings in source and target
      - E.g., “IS-1000” vs. “IS – 1000”
    - Special token sharing in translation and language model
      - One special token for each category: numbers (e.g., 2,596), math expressions (e.g., p<0.004), material names (e.g., C15H23N2O5P), and labeled names (e.g., 3.05g)
  - **Patent case-LM**
    - Retrain the case-LM on 45M LM data
    - Word segmentation lexicon
    - Re-optimize on 45M parallel corpus
    - Use only 100 features of the highest weights in each category of the 50K sparse features
    - Address over-fitting due to smaller tuning set
  - **Document-level CLIR-based LM adaptation**
    - Retrieve most relevant passages for a test document in the 14B LM data using CLIR
    - Bias LM for sentences in the test document to these passages

**Recent Advances**

- **Mistakes features**
  - Model target bigrams given source and vise versa
  - Trait features: model general properties of translation hypotheses, e.g., percent of words that re-order
- **Sentence-level LM adaptation instead of document-level**
  - Patent documents tend to use well-structured sentences and re-use n-grams in other patent documents
  - Extra LM data: 14B words of US patents in English
  - Extract all phrasal rules, ignoring dependency
  - **Bias LM for sentences in the test document to these passages**
  - Helpful in real patent examination and possibly other tasks

**Results**

- **On development set**
  - Development in NTCIR-9 with 45M LM
    - BLEU: 37.71
    - + translation-based true caser: 41.19
    - + recurrent neural network LM: 42.13
    - + translation-based true caser
    - + recurrent neural network LM
  - NTCIR-9 system with 14B LM
    - BLEU: 39.14
    - + translation-based true caser: 41.09
    - + recurrent neural network LM: 42.13
    - + translation-based true caser

- **Official evaluation**
  - Automatic Evaluation (BLEU)
    - System: BBN-1
      - BLEU: 39.98
    - BBN-2
      - BLEU: 42.68
    - Baseline1
      - BLEU: 36.69
    - Baseline2
      - BLEU: 36.95
      - + translation-based true caser: 41.09
    - + recurrent neural network LM: 42.13
    - + translation-based true caser

- **Typical example**
  - Source: 对于每一像素，变焦引擎210使用在以上等式(2)和(4)中的加权来确定所
    - 该像素是否在三角形内
  - MT output: For each pixel, the rendering engine 210 uses the edge equation set forth in equations (2) to (4) above to determine whether the pixel is in a triangle
  - Reference: For each pixel, the shading engine 210 determines whether the pixel is in the triangle using the edge equations set forth in equations (2) to (4) above.