

How do you pronounce your name? Improving G2P with transliterations

Aditya Bhargava and Grzegorz Kondrak
University of Alberta
ACL-HLT 2011

Introduction

- Name pronunciations can be fickle
 - Speech synthesis systems must handle them
 - Best G2P system can't account for how I *decide* my name is pronounced
- Existing transliterations encode this info
 - Ample data that can be easily mined from the Web

Objective: apply transliterations

Gershwin



/dʒɹʌʃwɪn/?

/ɡɹʌʃwɪn/?

...?

ガーシュウィン

Гершвин



Applying transliterations

- Assume existing G2P base systems
 - Produce n-best output lists
- Assume available transliteration
- Pick candidate output that is “most similar” to transliteration

Data

- G2P: Combilex
 - Provides “name” annotations
- Transliterations: NEWS Shared Task 2010
English-to-Hindi data
- Intersect data

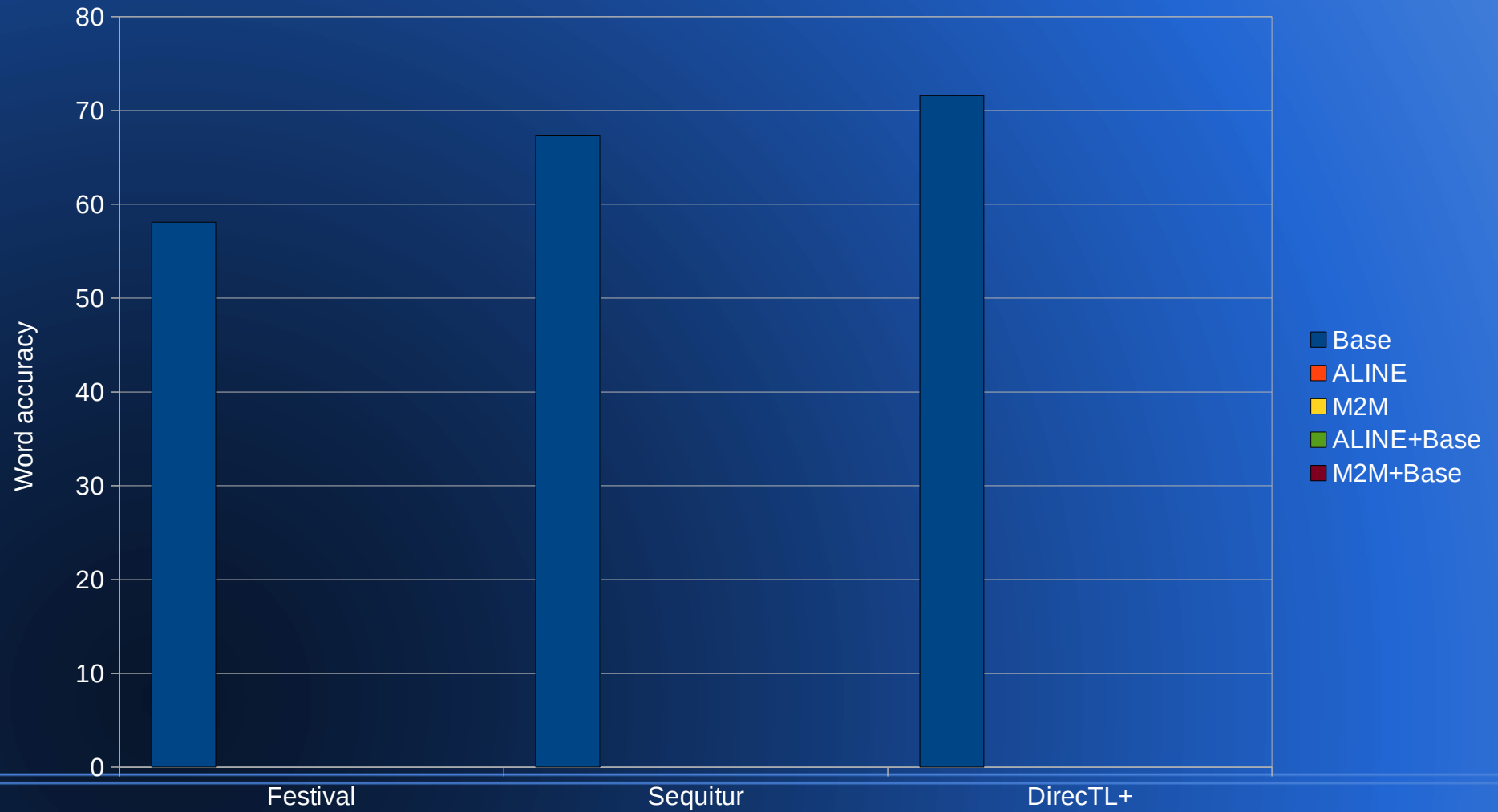
Base systems

- Festival (Black et al., 1998)
 - CARTs
 - Popular end-to-end speech synthesis
- Sequitur (Bisani and Ney, 2008)
 - Generative joint n-grams
 - G2P only
- DirecTL+ (Jiampojarn et al., 2008)
 - Discriminative phrasal decoding
 - G2P only

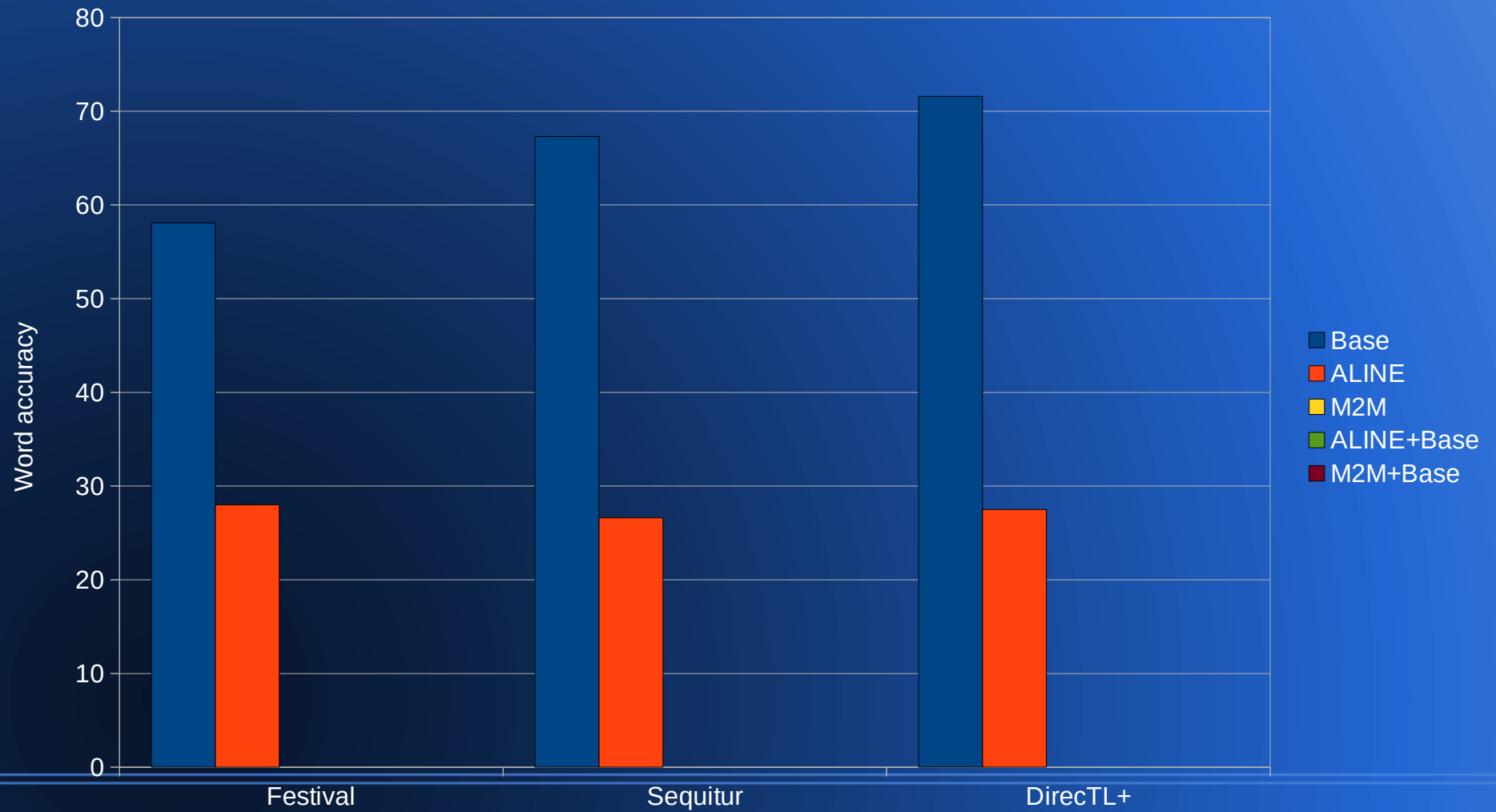
Similarity

- Similarity measures:
 - ALINE phoneme-to-phoneme aligner score
 - Rule-based G2P converter for Hindi
 - M2M-Aligner alignment system score
 - Extension of learned edit distance algorithm
- Two overall approaches:
 - Use highest similarity score
 - Combine similarity score with system score

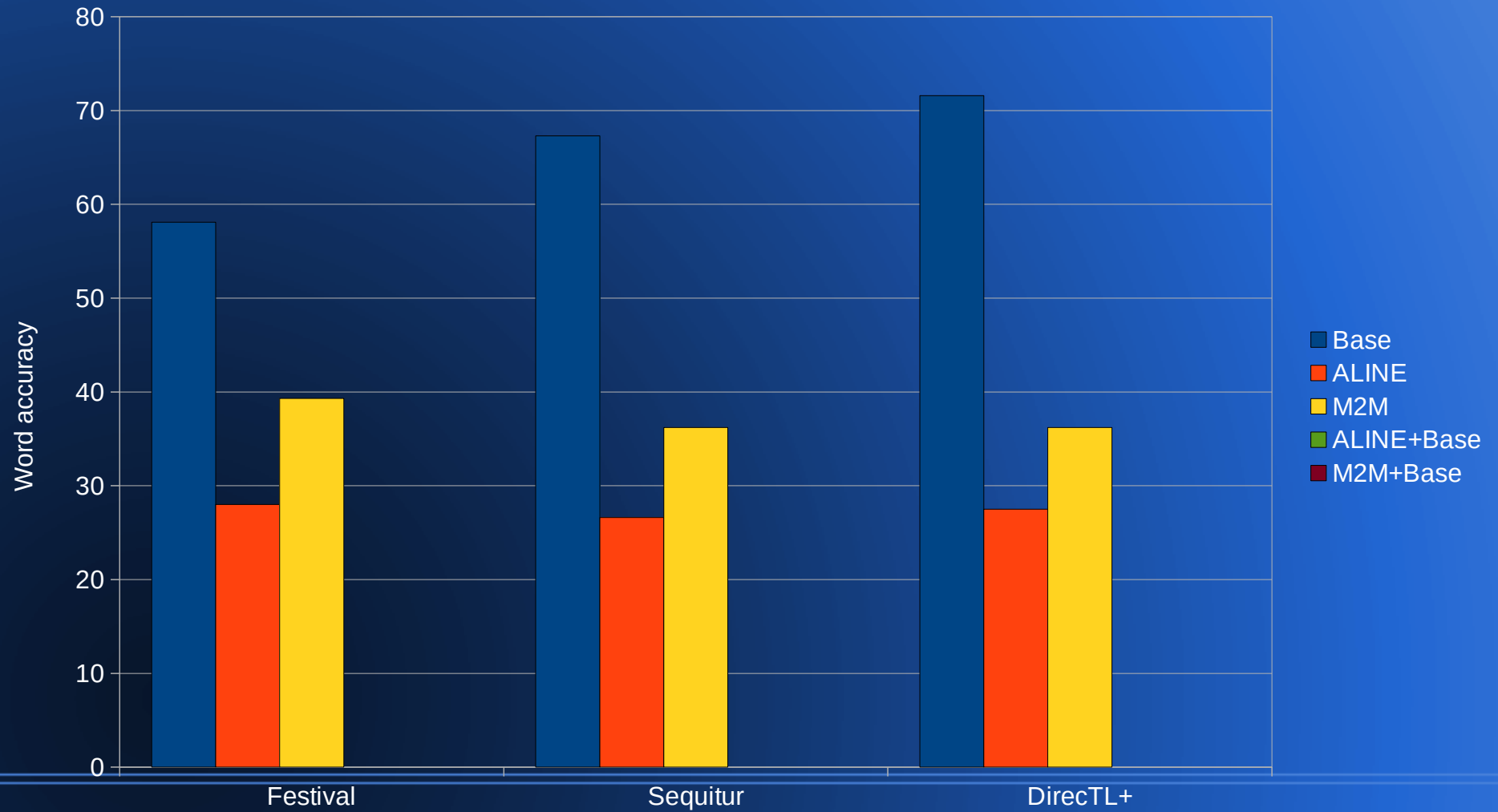
Similarity: results



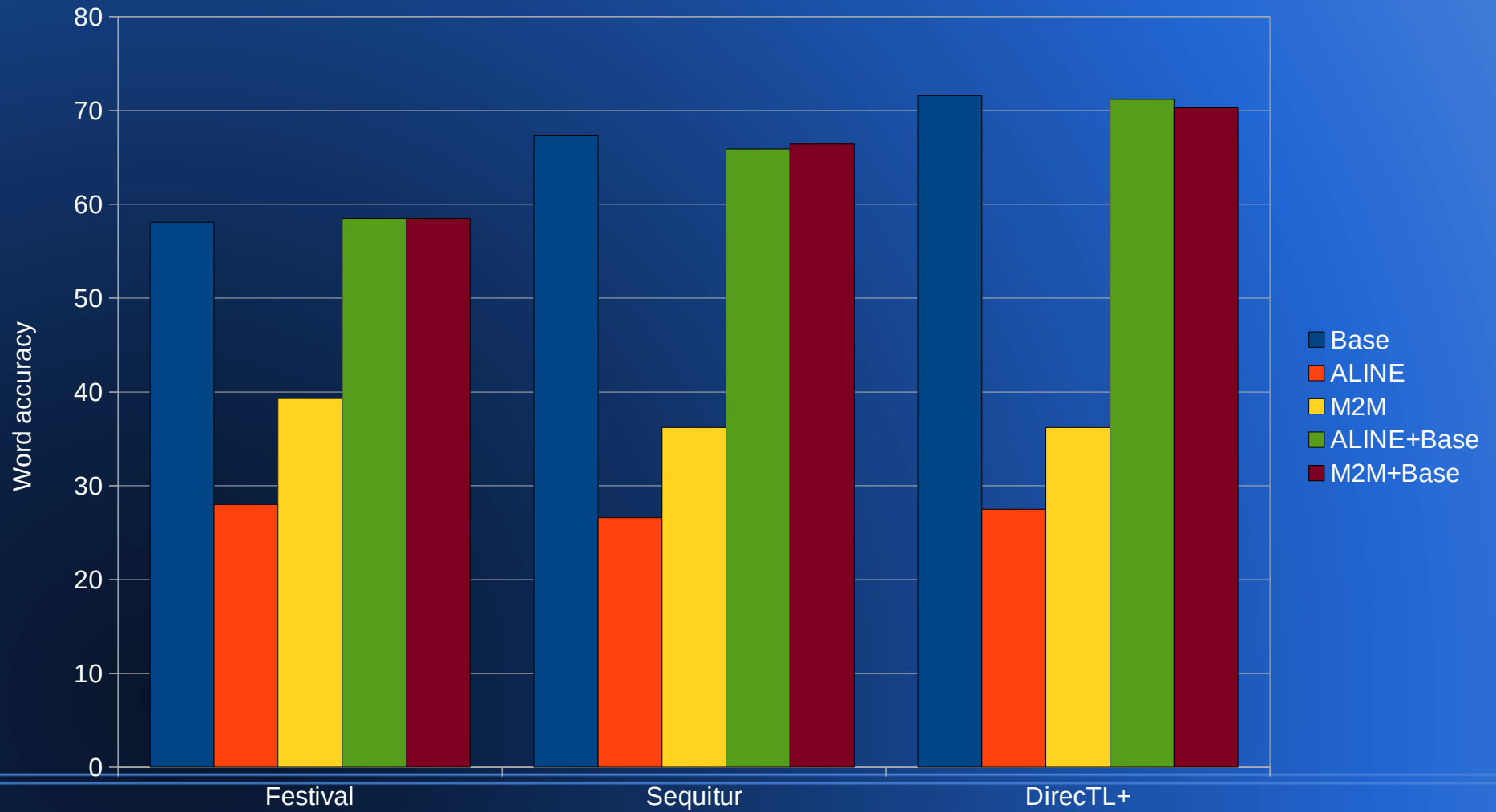
Similarity: results



Similarity: results



Similarity: results



Similarity: post mortem

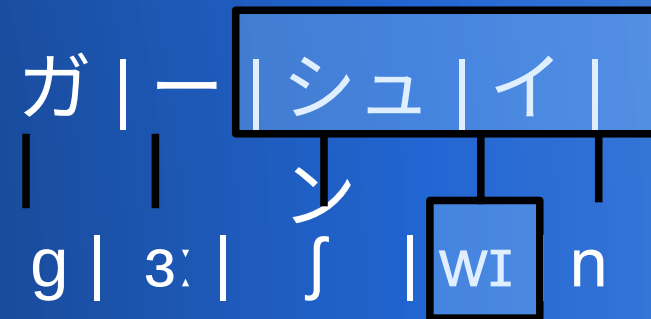
- Difficult to do!
- Can't follow transliterations exactly
 - Differences in scripts
 - Differences in languages (phonologies)
 - Noisy data
- Need to smooth out this volatility
- Limited to one language

SVM re-ranking

- Many features
 - Similarity scores (M2M-Aligner)
 - Score differences
 - N-grams based on alignments between transcriptions and transliterations
 - Similar to features used in DirecTL+

SVM re-ranking

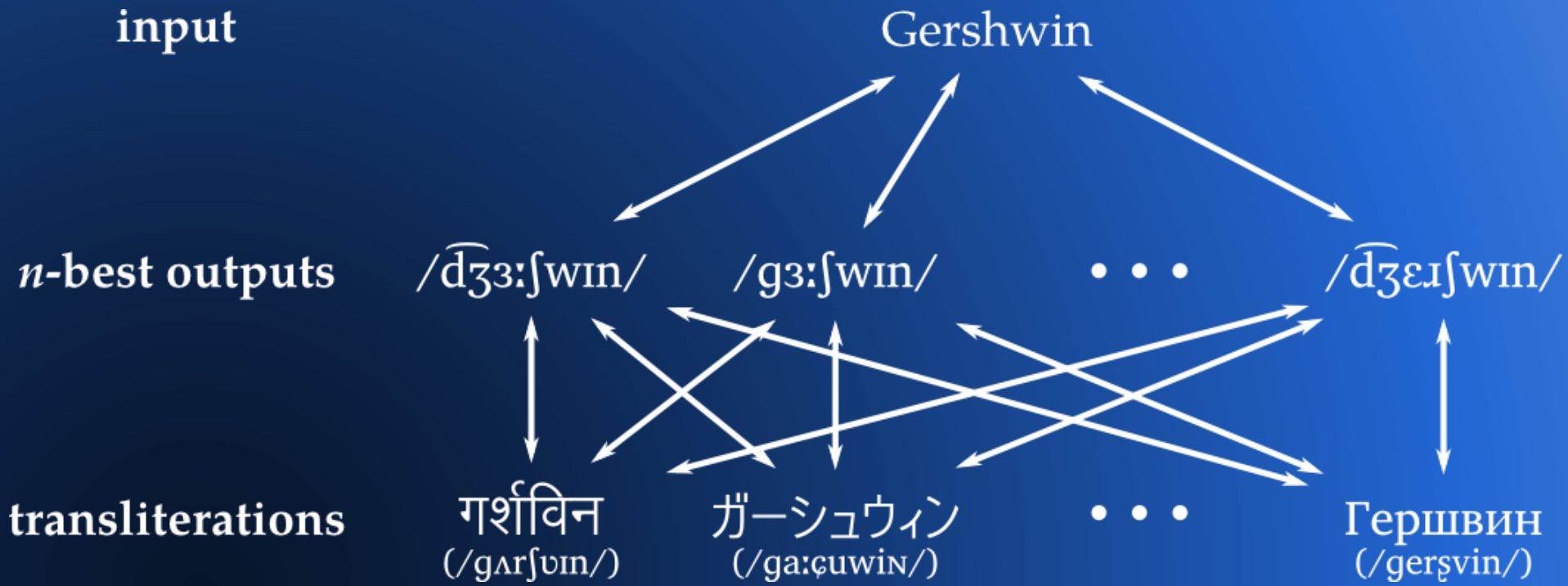
- Many features
 - Similarity scores (M2M-Aligner)
 - Score differences
 - N-grams based on alignments between transcriptions and transliterations
 - Similar to features used in DirecTL+



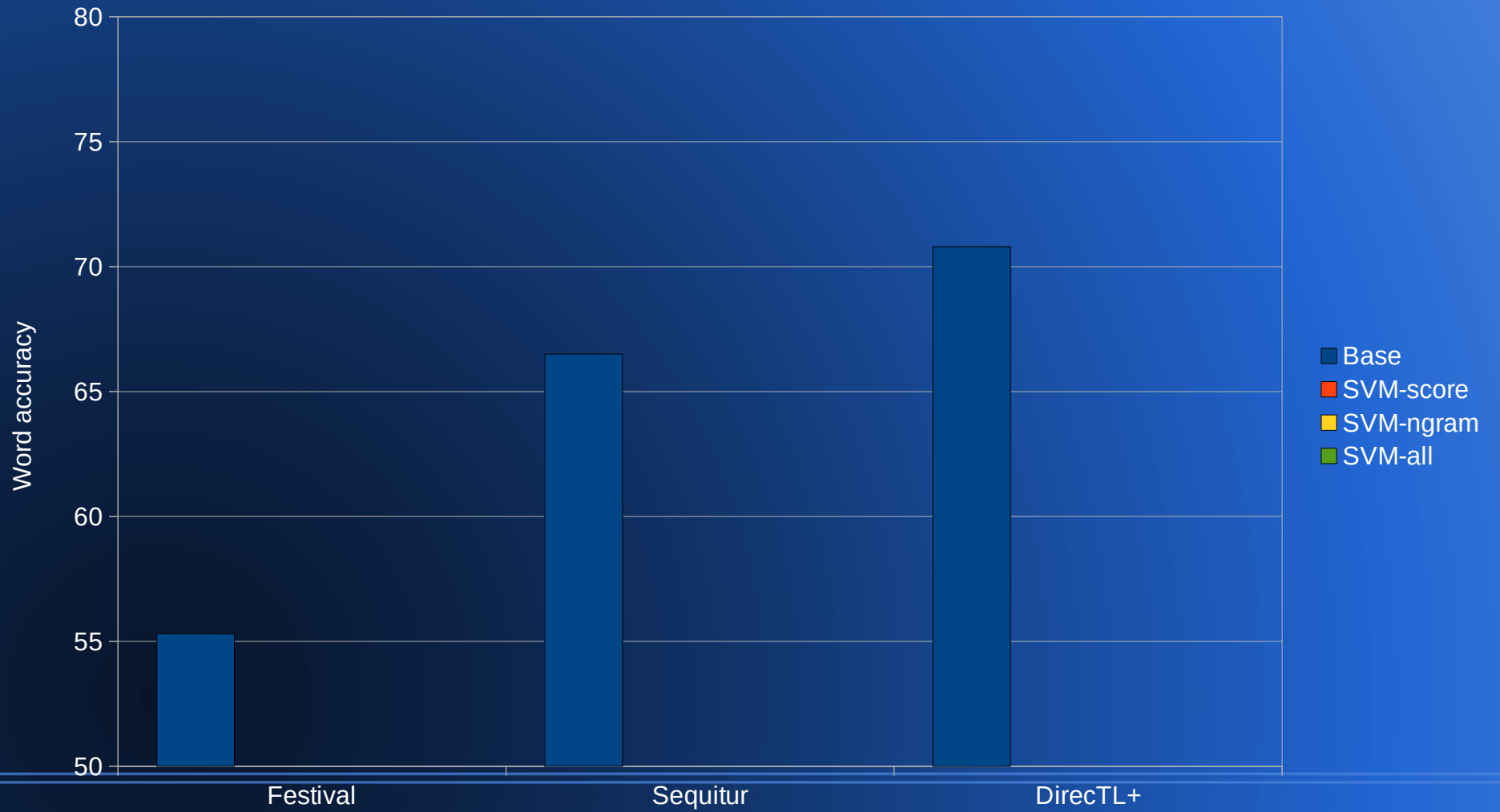
SVM re-ranking

- Allows many languages
 - English-to-{Bengali, Chinese, Hindi, Thai, Japanese, Kannada, Korean, Russian, Tamil}
 - Features repeated for each transliteration

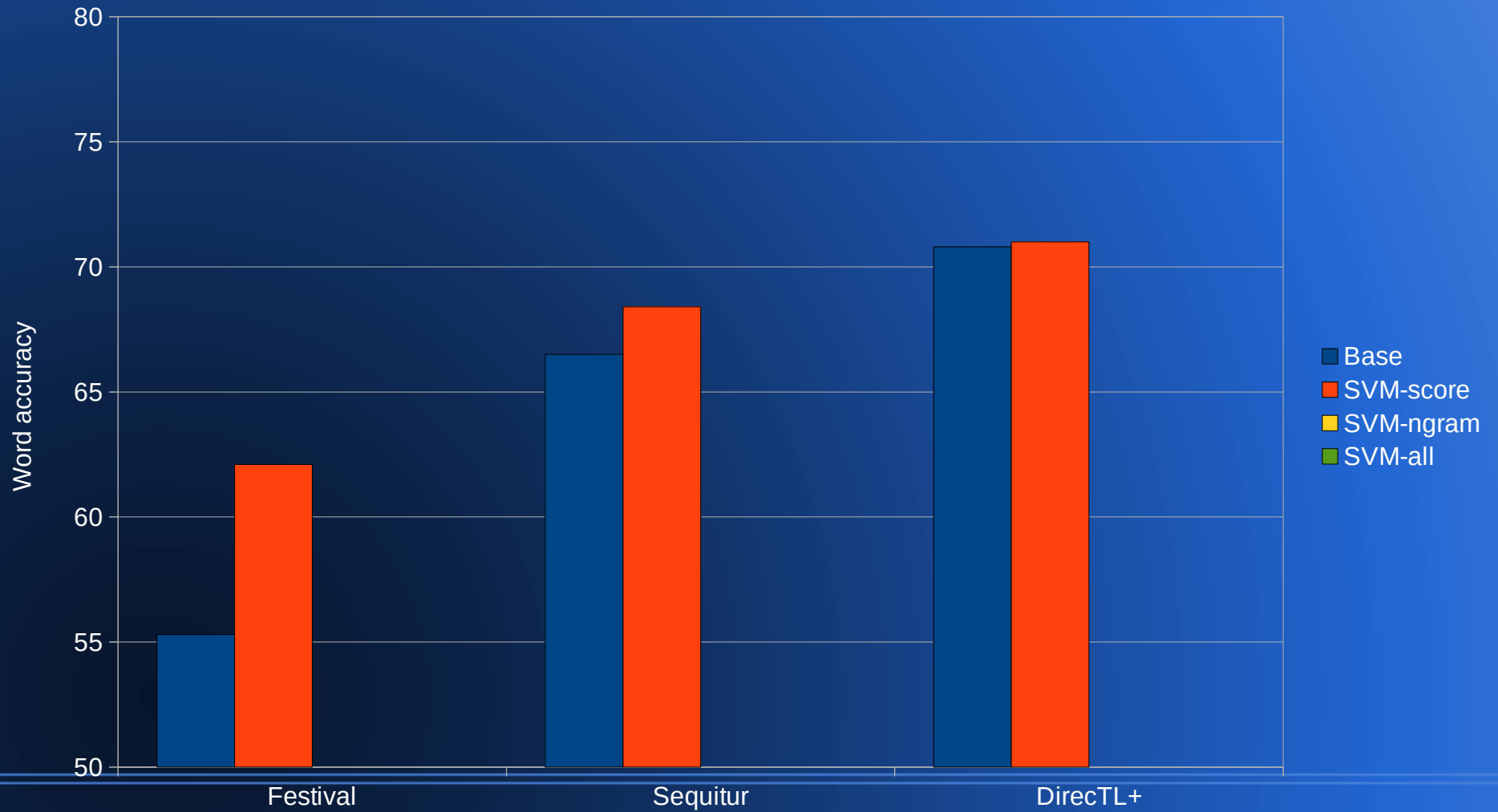
SVM re-ranking



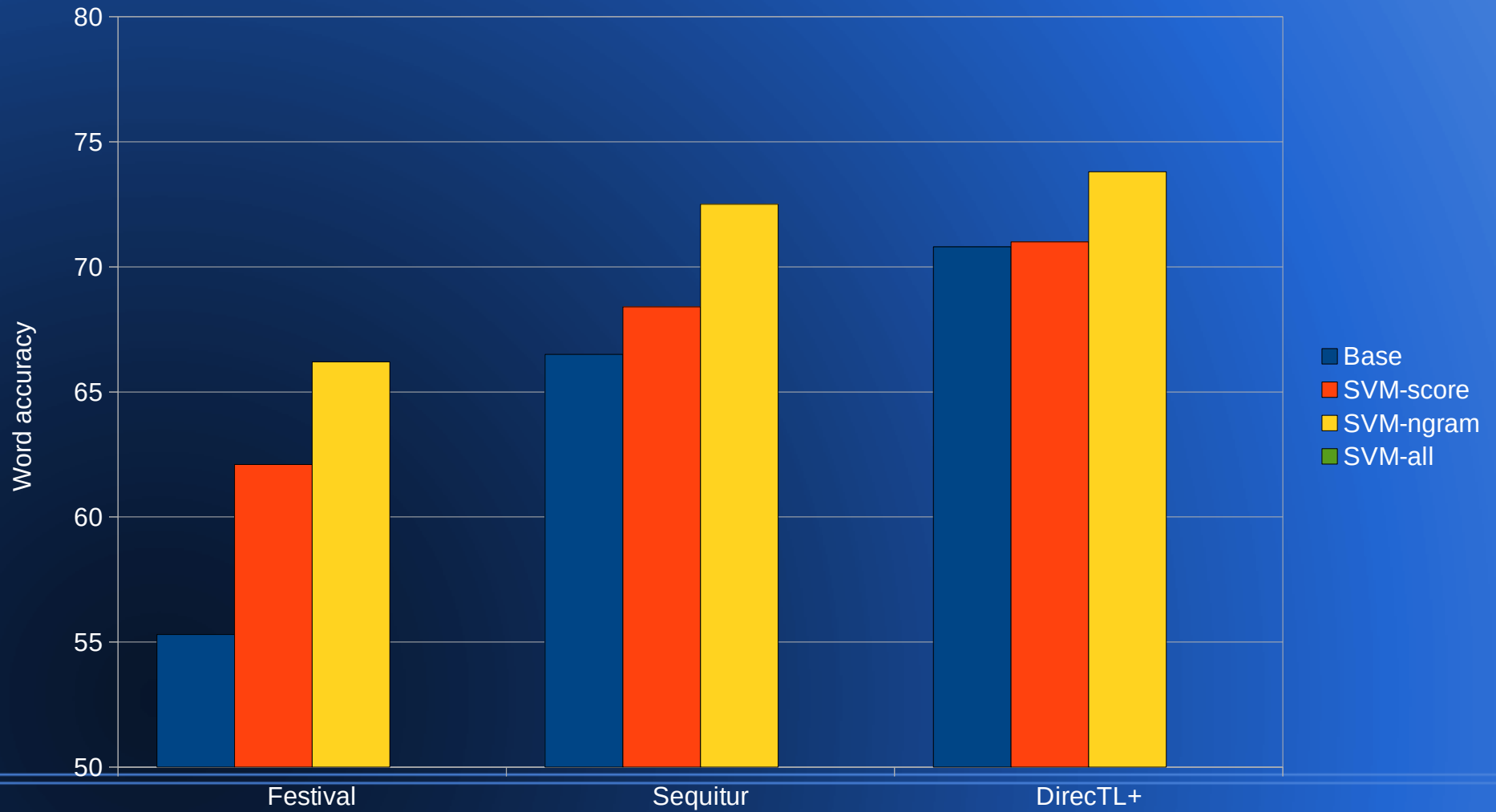
SVM re-ranking



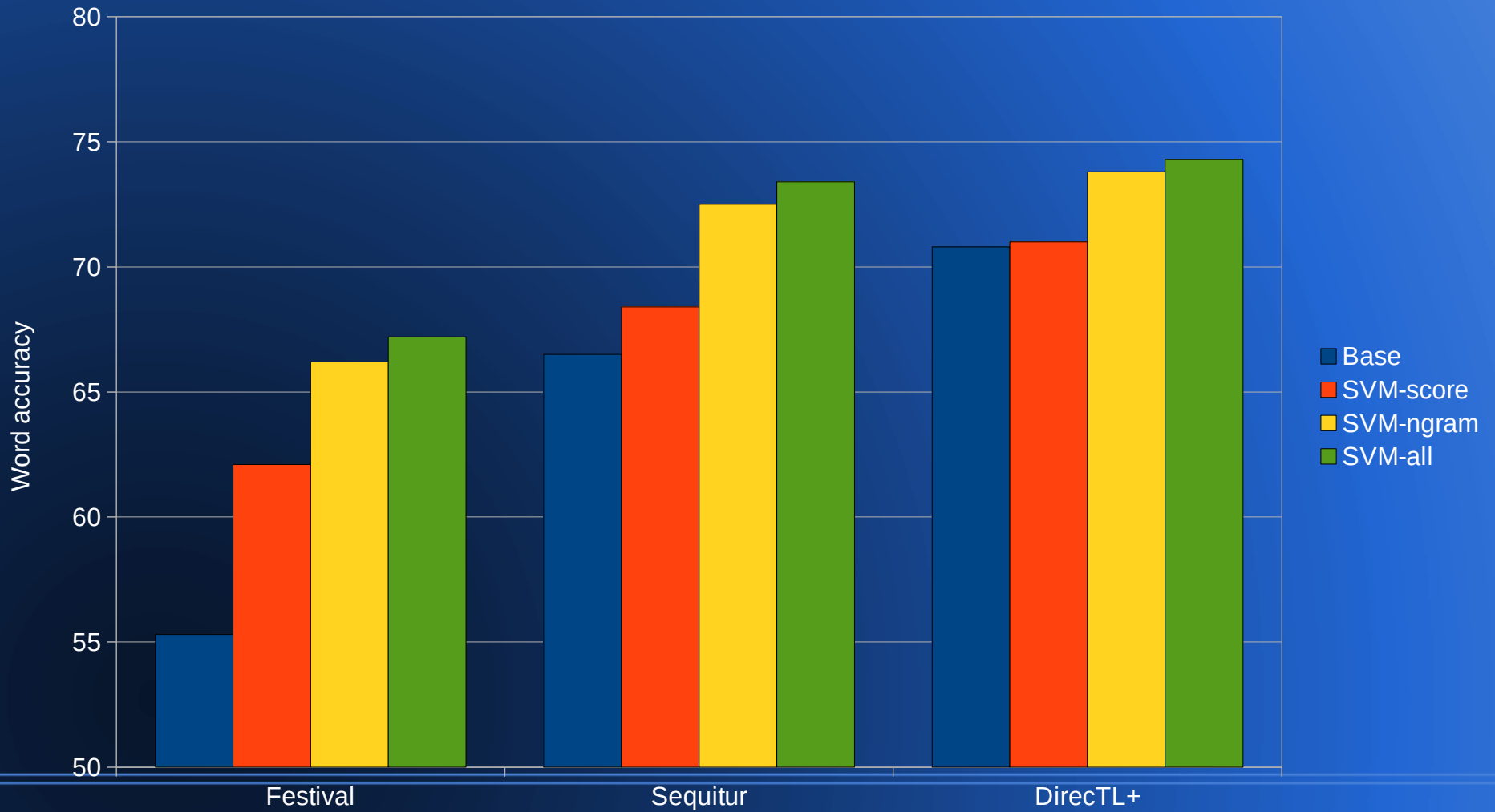
SVM re-ranking



SVM re-ranking



SVM re-ranking

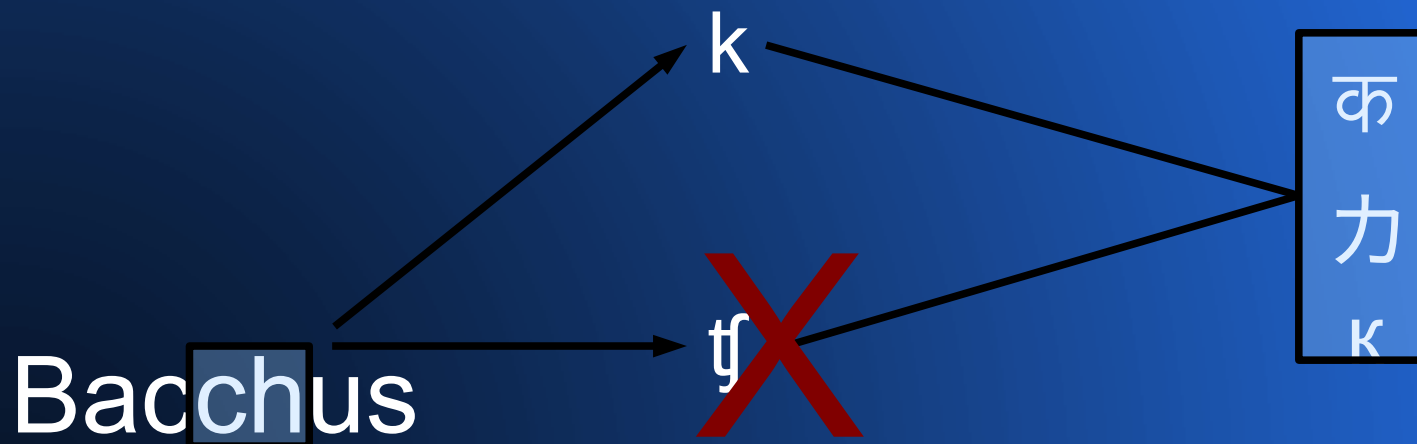


Analysis

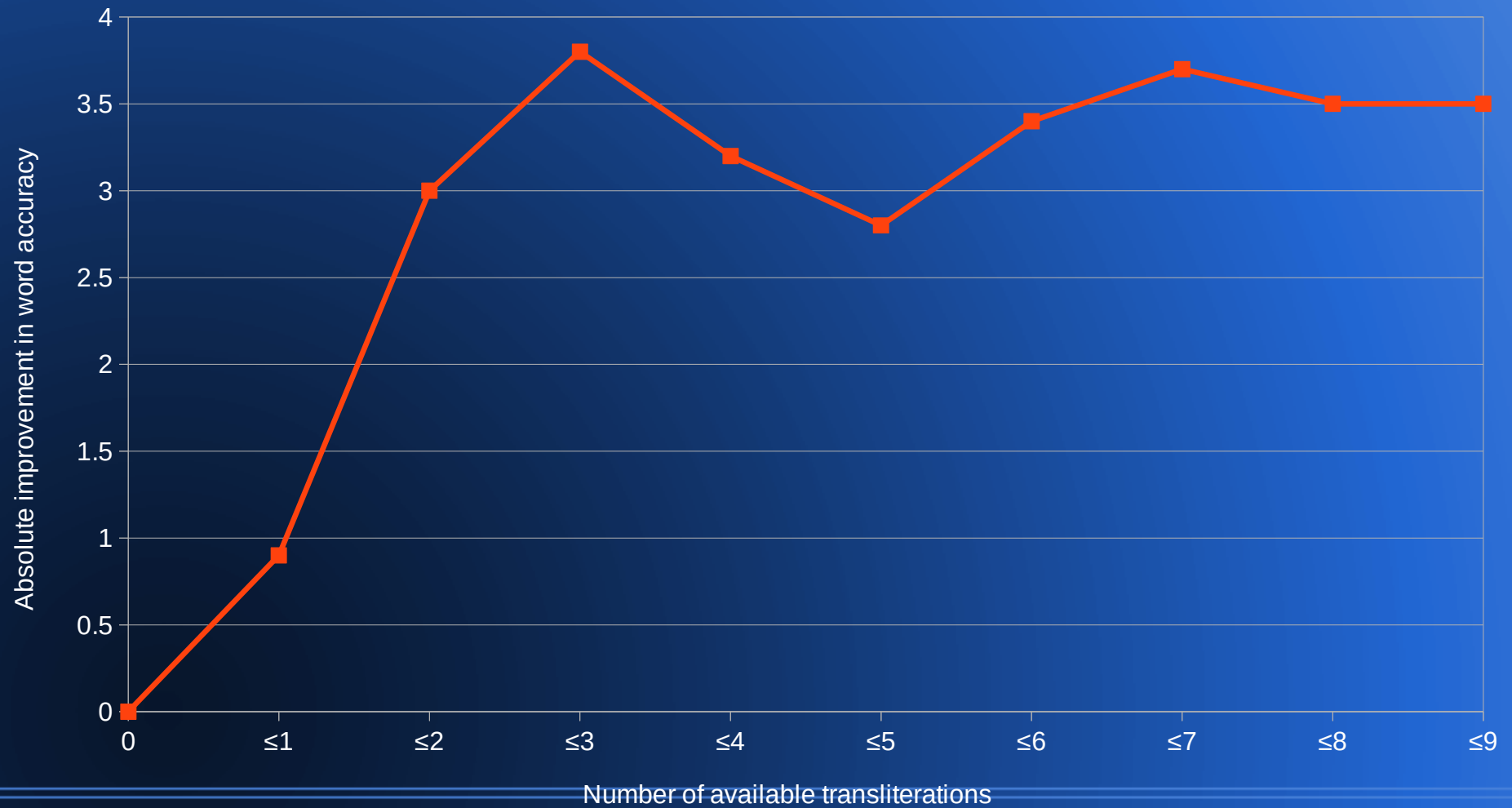
- SVM re-ranking gives significant improvements
- Festival and Sequitur get higher improvement
 - The better the base system, the harder it is to re-rank
 - *n*-gram features styled after DirecTL+
 - This benefits Festival and Sequitur
- **Similar features in a novel direction** can lead to improved performance

Analysis

- N-gram features most useful
 - Granular features
 - Includes unable-to-align feature



Multiple languages



Future work

- Apply same re-ranking approach to different tasks (e.g. transliteration) and different data (e.g. transcriptions)
 - Very successful results so far
- Leverage noisy web transcriptions
- Incorporate supplemental information directly in system

Conclusion

- First use of transliterations for G2P
- Basic similarity-based methods don't work
- SVM re-ranking improves all tested base systems
- Multiple languages are vital
- Relevant scripts, etc. are online