UNIVERSIDAD POLITÉCNICA DE VALNECIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

SIMILARITY WORD-SEQUENCE KERNELS FOR SENTENCE CLUSTERING

Jesús Andrés-Ferrer, G. SANCHÍS-TRILLES AND F. CASACUBERTA
{jandres,gsanchis,fcn}@disc.upv.es

Contents

1 Introduction

- ***Text classification***: classify a given document or text $\mathbf{x}$ into a class c from a ***known*** set of classes
  - ***Text clustering***: the set of classes is ***unknown***
    - Sentence clustering: each document or text is composed by one sentence
    - Bilingual sentence clustering: the same sentence in two different languages
- Motivation [for (bilingual) sentence clustering]:
  - Training specific models
    - Domain adaptation
    - Reduction in time complexity
- Properties of clustering:
  - It is a NP-Hard problem
  - A distance between objects (documents) is needed $\mathrm{d}(\mathbf{x}, \mathbf{x}')$
- Lloyd's algorithm or C-means is a fast and sub-optimal algorithm
  - It is unable to find suitable clusters whenever the given data are not linearly separable

- Some works proposed an extension of C-means that relies on Mercer Kernels
  - Map the objects $\mathbf{x}$ and $\mathbf{x}'$ into a higher dimensionality domain in which can be linearly separable

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}'), \tag{1}$$

  - $\phi(\mathbf{x})$ is the mapping function to a higher-dimensionality feature space

- Since Kernels are symmetric, some could be used as similarity (or distance) functions: $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$

- We present a clustering algorithm that uses kernels as similarity functions

2 C-means clustering

- The minimization of a ***“distance”*** is a common criterion for clustering:
  - Given a set of samples: $\{\mathbf{x}_n\}_1^N$ and a number of clusters C
  - Find the set of index variables $\{\mathbf{z}_n\}$ that minimize:

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \left\{ \frac{1}{N} \sum_{c=1}^{C} \sum_{n=1}^{N} z_{nc}\, \mathrm{d}(\mathbf{x}_n, \mathbf{m}_c) \right\}, \tag{2}$$

  - with:
    - $\mathbf{m}_c = \frac{1}{N_c} \sum_{n=1}^{N} z_{nc} \mathbf{x}_n$
    - $N_c = \sum_{n=1}^{N} z_{nc}$
    - $z_{nc} = \begin{cases} 1 & \text{if } \mathbf{x}_n \text{ belongs to the } c\text{-th cluster} \\ 0 & otherwise \end{cases}$
- The C-means algorithm seeks to find a local minimum for the ***2-norm***:

$$\mathrm{d}(\mathbf{x}_n, \mathbf{m}_c) = (\mathbf{x}_n - \mathbf{m}_c)^T (\mathbf{x}_n - \mathbf{m}_c). \tag{3}$$

- The distance used by the C-means algorithm can either be a *pseudo-metric* or a *semi-metric*

2.1 Kernel-based C-means clustering

- C-means can be extended with Mercer Kernels:
  - Change the distance function by:

$$\mathrm{d}(\mathbf{x}_n, \mathbf{m}_c) = (\phi(\mathbf{x}_n) - \mathbf{m}_c)^T (\phi(\mathbf{x}_n) - \mathbf{m}_c), \quad (4)$$

  - with $\mathbf{m}_c = \frac{1}{N_c} \sum_{n=1}^{N} \mathbf{z}_{nc} \phi(\mathbf{x}_n)$
- Kernels verify the ***symmetric*** requirement to be a ***pseudo-metric***, additional requirements:
  - ***Positiveness***
- For being a ***semi-metric***:
  - ***pseudo-metric***
  - ***Identity of indiscernibles***
- For being a ***metric***:
  - ***semi-metric***
  - ***Triangle inequality***
- Kernels are more naturally redefined as similarity functions
- Given a distance, a similarity can be defined and vice-versa.

2.2 Similarity Kernel-based C-means clustering

- Kernels are more naturally redefined as similarity functions

- Given a distance, a similarity can be defined and vice-versa

- C-means can be re-defined in terms of similarity:

$$\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} \left\{ \frac{1}{N} \sum_{c=1}^{C} \sum_{n=1}^{N} z_{nc} \, \mathrm{s}(\mathbf{x}_n, \mathbf{m}_c) \right\}, \tag{5}$$

- with:
  - $\mathbf{m}_c = \frac{1}{N_c} \sum_{n=1}^{N} z_{nc} \phi(\mathbf{x}_{nc})$,
  - $\mathrm{s}(\mathbf{x}_n, \mathbf{m}_c) = \phi(\mathbf{x}_{nc})^T \mathbf{m}_c$

- We propose several similarity kernels for text clustering

3 Word-sequence kernels (WSK)

- Compute strings similarity based on matching (non-)consecutive sequences of symbols

- Define a mapping: $\Sigma^n \rightarrow \mathbb{R}^{|\Sigma|^n}$,

- where:
  - n : the maximum length of the segment to be considered

- For a given order n and a pair of documents $\mathbf{x}$, and $\mathbf{x}'$:

$$K_n(\mathbf{x}, \mathbf{x}') = \sum_{u \in \Sigma^n} |\mathbf{x}|_u |\mathbf{x}'|_u, \quad (6)$$

- where $|\mathbf{x}|_u$ is the number of occurrences of u in document $\mathbf{x}$

- Neither it is a semi-similarity, nor a pseudo-similarity

3.1 0-1 WSK

- We define the kernel K_n^1 as follows:

$$K_n^1(\mathbf{x}, \mathbf{x}') = \sum_{u \in \Sigma^n} 1_u(\mathbf{x}) 1_u(\mathbf{x}'), \tag{7}$$

- with $1_u(\mathbf{x}) = \begin{cases} 1 & \text{if } u \text{ occurs in } \mathbf{x} \\ 0 & \text{otherwise} \end{cases}$

- It is not a semi-similarity

- It is a pseudo-similarity

- It behave like a semi-similarity in practice

3.2 Normalized WSK

- We can normalize the both kernels, WSK and 0–1 WSK
  - WSK:

$$\hat{K}_n(\mathbf{x}, \mathbf{x}') = \sum_{u \in \Sigma^n} \frac{|\mathbf{x}|_u}{\sqrt{\sum_{v \in \Sigma^n} |\mathbf{x}|_v}} \frac{|\mathbf{x}'|_u}{\sqrt{\sum_{v \in \Sigma^n} |\mathbf{x}'|_v}} \tag{8}$$

    - It is not a semi-similarity
  - 0–1 WSK:

$$\hat{K}_n^1 = \sum_{u \in \Sigma^n} \frac{1_u(\mathbf{x})}{\sqrt{\sum_{v \in \Sigma^n} 1_v(\mathbf{x})}} \frac{1_u(\mathbf{x}')}{\sqrt{\sum_{v \in \Sigma^n} 1_v(\mathbf{x}')}} \tag{9}$$

    - It is a semi-similarity

3.3 Sum WSK

- n-grams are very sparse for large values of n

- $\bar{K}_n$ is defined as

$$\bar{K}_n(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{n} \hat{K}_i(\mathbf{x}, \mathbf{x}'). \tag{10}$$

- $\bar{K}_n^1$ is defined as

$$\bar{K}_n^1(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{n} \hat{K}_i^1(\mathbf{x}, \mathbf{x}'). \tag{11}$$

3.4 Examples

- Consider the following 4 strings:

$$\mathbf{s}_1 = \{abcb\} \qquad \mathbf{s}_2 = \{abab\}$$
$$\mathbf{s}_3 = \{abeb\} \qquad \mathbf{s}_4 = \{abcbab\}$$

- “$\mathbf{s}_1$ is as similar to $\mathbf{s}_2$ as to $\mathbf{s}_3$” (Assuming Levenshtein distance)

3.5 Examples

- Consider the following 4 strings:

$$\mathbf{s}_1 = \{abcb\} \qquad \mathbf{s}_2 = \{abab\}$$
$$\mathbf{s}_3 = \{abeb\} \qquad \mathbf{s}_4 = \{abcbab\}$$

- "$\mathbf{s}_1$ is as similar to $\mathbf{s}_2$ as to $\mathbf{s}_3$" (Assuming Levenshtein distance)
- Analise $K_2(\ldots)$
  - $K_2(\mathbf{s}_1, \mathbf{s}_2) = 2$ and $K_2(\mathbf{s}_1, \mathbf{s}_3) = 1$
  - $K_2(\mathbf{s}_1, \mathbf{s}_4) = 4 > K_2(\mathbf{s}_1, \mathbf{s}_1) = 3$

3.6 Examples

- Consider the following 4 strings:

$$\mathbf{s}_1 = \{abcb\} \qquad \mathbf{s}_2 = \{abab\}$$
$$\mathbf{s}_3 = \{abeb\} \qquad \mathbf{s}_4 = \{abcbab\}$$

- "$\mathbf{s}_1$ is as similar to $\mathbf{s}_2$ as to $\mathbf{s}_3$" (Assuming Levenshtein distance)
- Analise $K_2(\ldots)$
  - $K_2(\mathbf{s}_1, \mathbf{s}_2) = 2$ and $K_2(\mathbf{s}_1, \mathbf{s}_3) = 1$
  - $K_2(\mathbf{s}_1, \mathbf{s}_4) = 4 > K_2(\mathbf{s}_1, \mathbf{s}_1) = 3$
- Analise $K_2^1(\ldots)$
  - $K_2^1(\mathbf{s}_1, \mathbf{s}_2) = 1$ and $K_2^1(\mathbf{s}_1, \mathbf{s}_3) = 1$
  - $K_2^1(\mathbf{s}_1, \mathbf{s}_1) = 3$ and $K_2^1(\mathbf{s}_1, \mathbf{s}_4) = 3$

3.7 Examples

- Consider the following 4 strings:

$$\mathbf{s}_1 = \{abcb\} \qquad \mathbf{s}_2 = \{abab\}$$
$$\mathbf{s}_3 = \{abeb\} \qquad \mathbf{s}_4 = \{abcbab\}$$

- "$\mathbf{s}_1$ is as similar to $\mathbf{s}_2$ as to $\mathbf{s}_3$" (Assuming Levenshtein distance)
- Analise $K_2(\ldots)$
  - $K_2(\mathbf{s}_1, \mathbf{s}_2) = 2$ and $K_2(\mathbf{s}_1, \mathbf{s}_3) = 1$
  - $K_2(\mathbf{s}_1, \mathbf{s}_4) = 4 > K_2(\mathbf{s}_1, \mathbf{s}_1) = 3$
- Analise $K_2^1(\ldots)$
  - $K_2^1(\mathbf{s}_1, \mathbf{s}_2) = 1$ and $K_2^1(\mathbf{s}_1, \mathbf{s}_3) = 1$
  - $K_2^1(\mathbf{s}_1, \mathbf{s}_1) = 3$ and $K_2^1(\mathbf{s}_1, \mathbf{s}_4) = 3$
- Analise $\hat{K}_2^1(\ldots)$
  - $\hat{K}_2^1(\mathbf{s}_1, \mathbf{s}_1) = 1$ which is larger than $\hat{K}_2^1(\mathbf{s}_1, \mathbf{s}_4) = 0.866$
  - Identity of indiscernibles, a required property to assure C-means convergence
- The Kernel $\hat{K}_2(\ldots)$ reduces the cases for which it is not a semi-metric

4 Bilingual word-sequence kernels (BWSK)

- Previous WSK can be extended to bilingual documents:
  - $\mathbf{w} = \{\mathbf{x}, \mathbf{y}\}$ a bilingual sentence pair
    - $\mathbf{x}$ is a source sentence
    - $\mathbf{y}$ is a target sentence [a translation of source sentence]
  - Define the mapping: $\Sigma \times \Delta \rightarrow \mathbb{R}^{|\Sigma|^n} \times \mathbb{R}^{|\Delta|^n}$:

$$B_n(\mathbf{w}, \mathbf{w}') = K_n(\mathbf{x}, \mathbf{x}') + K_n(\mathbf{y}, \mathbf{y}') = \sum_{u \in \Sigma^n} |\mathbf{x}|_u |\mathbf{x}'|_u + \sum_{v \in \Delta^n} |\mathbf{y}|_v |\mathbf{y}'|_v \quad (12)$$

  - Similarly the following kernels are defined:
    - $B_n^1(\mathbf{w}, \mathbf{w}')$
    - $\hat{B}_n^1(\mathbf{w}, \mathbf{w}')$
    - $\bar{B}_n^1(\mathbf{w}, \mathbf{w}')$
    - $\hat{B}_n(\mathbf{w}, \mathbf{w}')$
    - $\bar{B}_n(\mathbf{w}, \mathbf{w}')$

5 Experiments

5.1 Corpora

- 2 corpora were used:
  - BTEC (Basic Travel Expression Corpus) [Chinese-English]

Language	N. Sentences	Running words	Vocabulary	Perplexity
Chinese	20K	172K	8428	24.3
English	20K	183K	7298	20.8

  - Europarlv3 with sentence length smaller or equal to 20 [Spanish-English]

Language	N. Sentences	Running words	Vocabulary	perplexity
Spanish	312K	4.0M	58K	28.2
English	312K	3.9M	37K	26.7

- All singletons were filtered out from training data [No effect]

- Stop-words were also filtered

5.2 Evaluation metric

- Typically, average intra-cluster distance/similarity is used to asses cluster quality

- C-means minimizes/maximizes these measures, so they are always improved

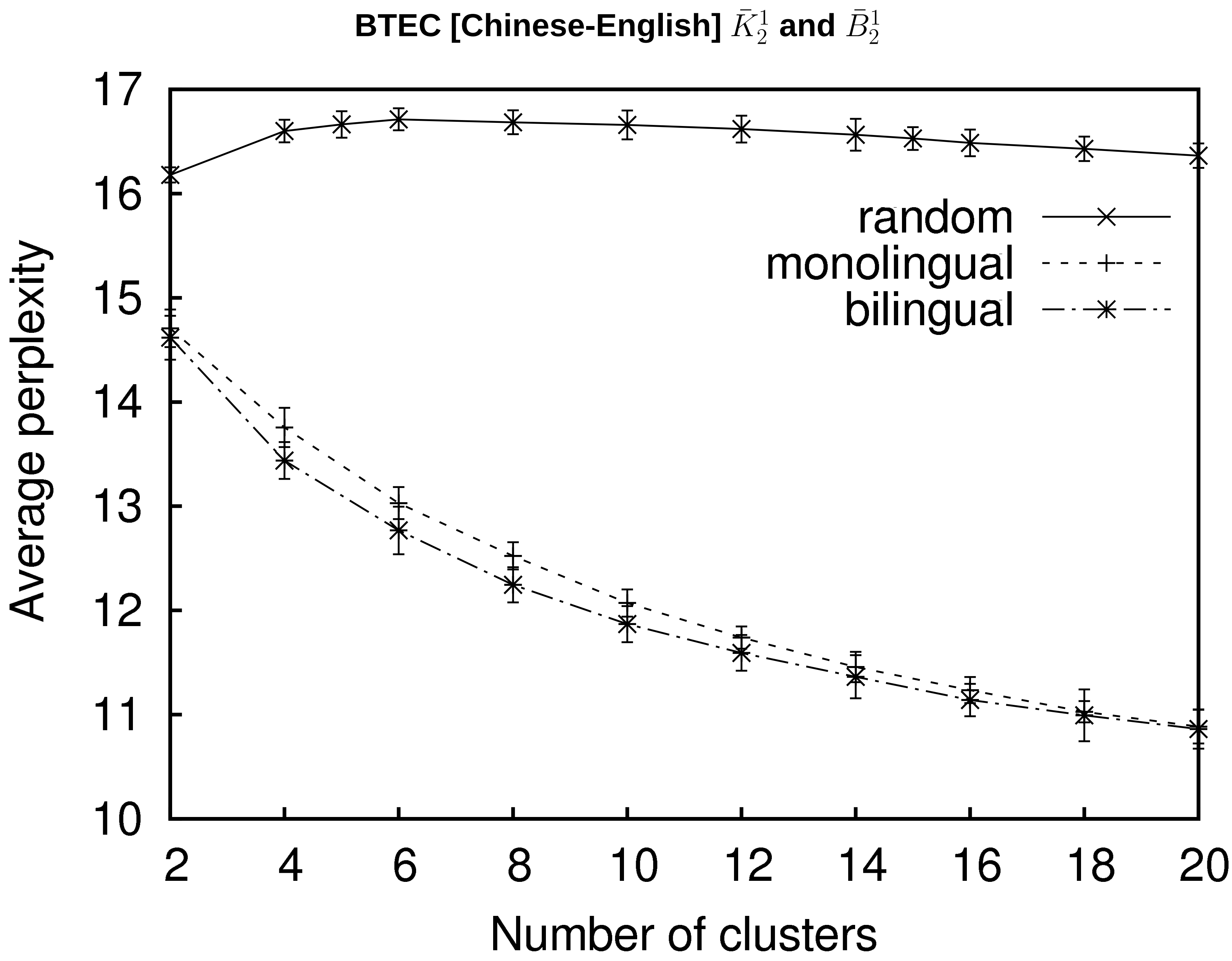
- 2 alternative measures:
  - Intra-cluster perplexity (IC-PPL) average:

$$ppl_{avg} = 2^{\sum_{c=1}^{C} \frac{1}{C} \frac{1}{W_c} \log_2 p(c)}, \tag{13}$$

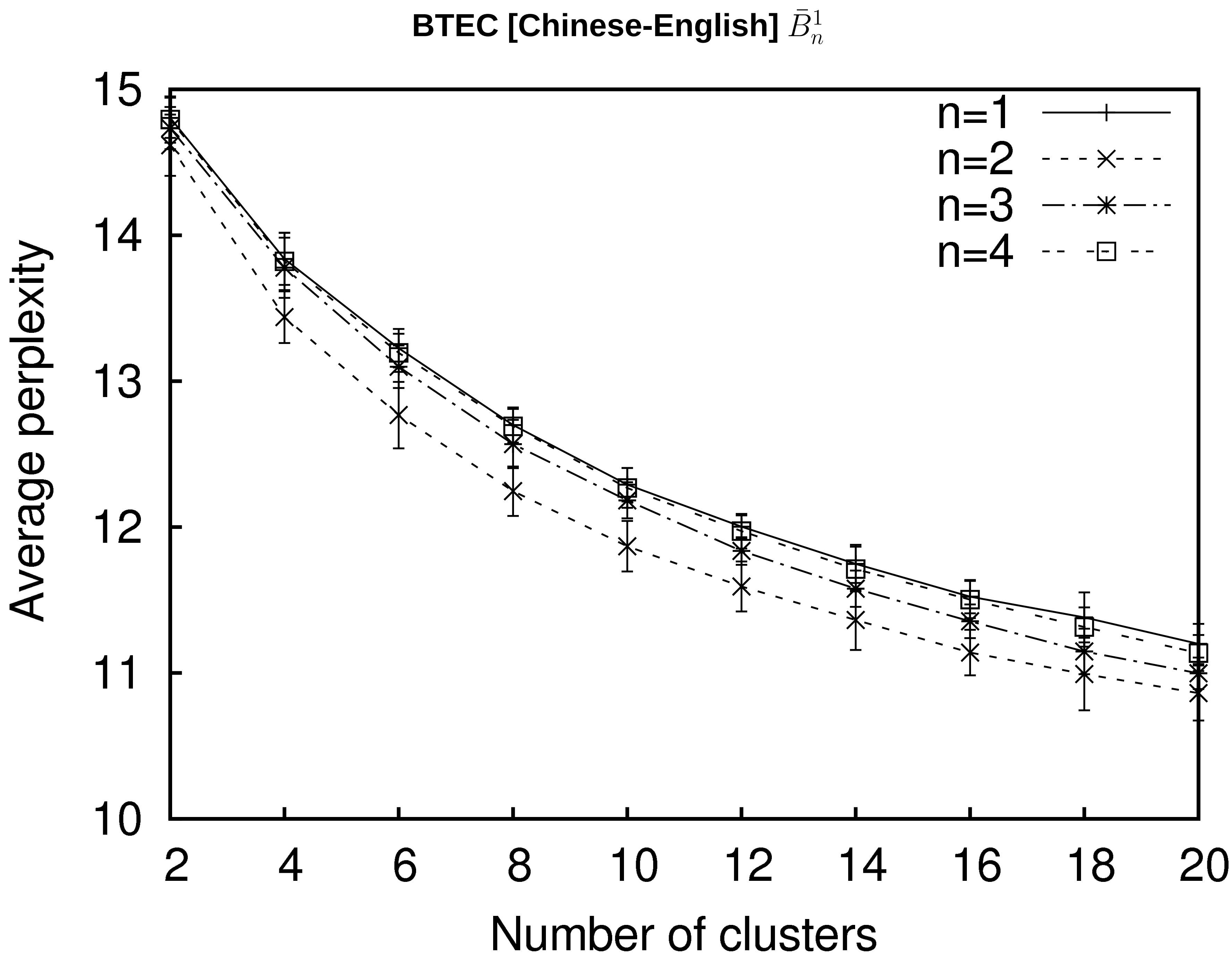
    - with $p(c)$ is the probability of the samples of cluster c according to the language model estimated on that same cluster
  - Edit distance [Equivalent in practice]

- IC-PPL for 5-grams in the English part is used through the experiments

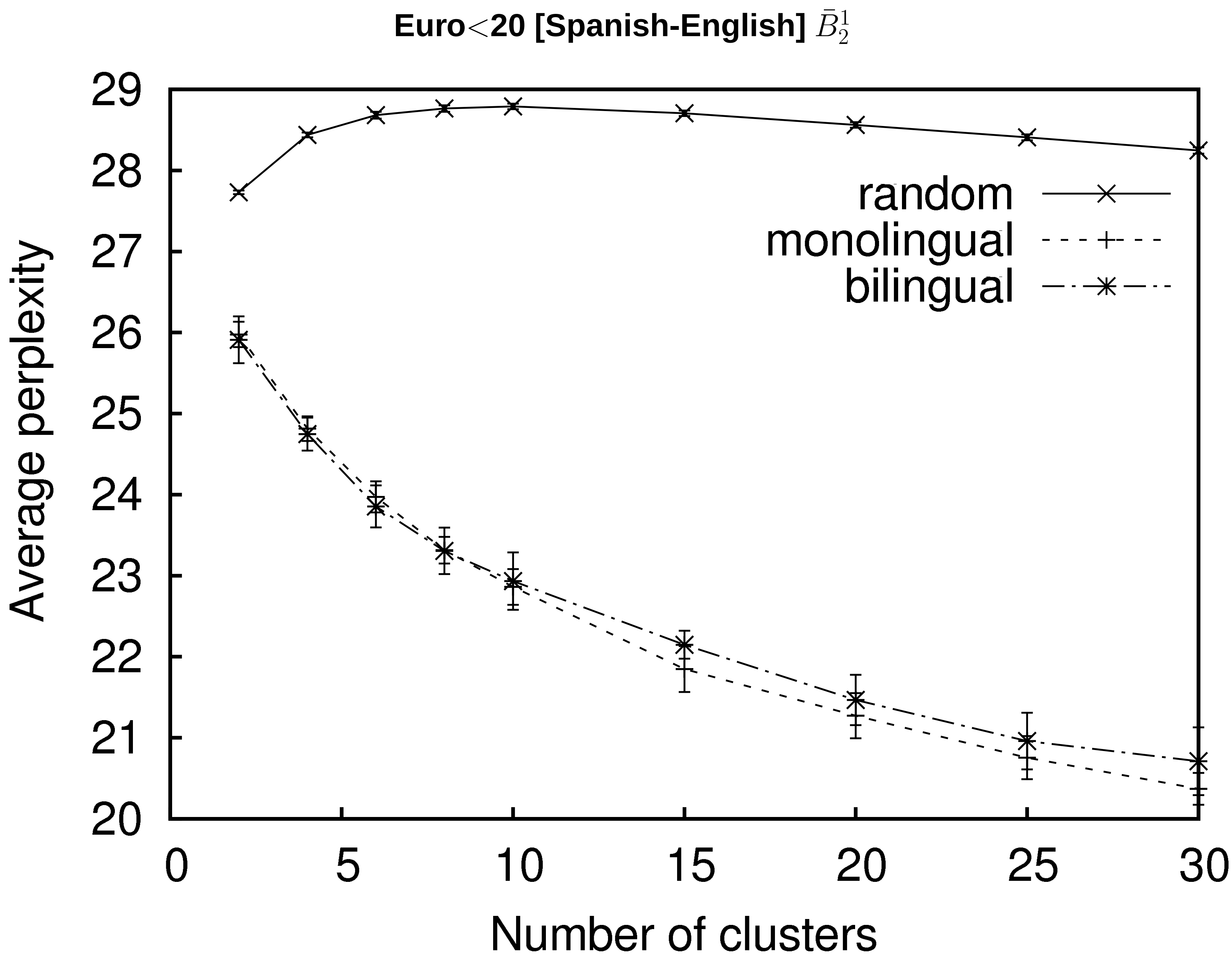
- LM where smoothed with the interpolated modified Kneser-Ney smoothing technique

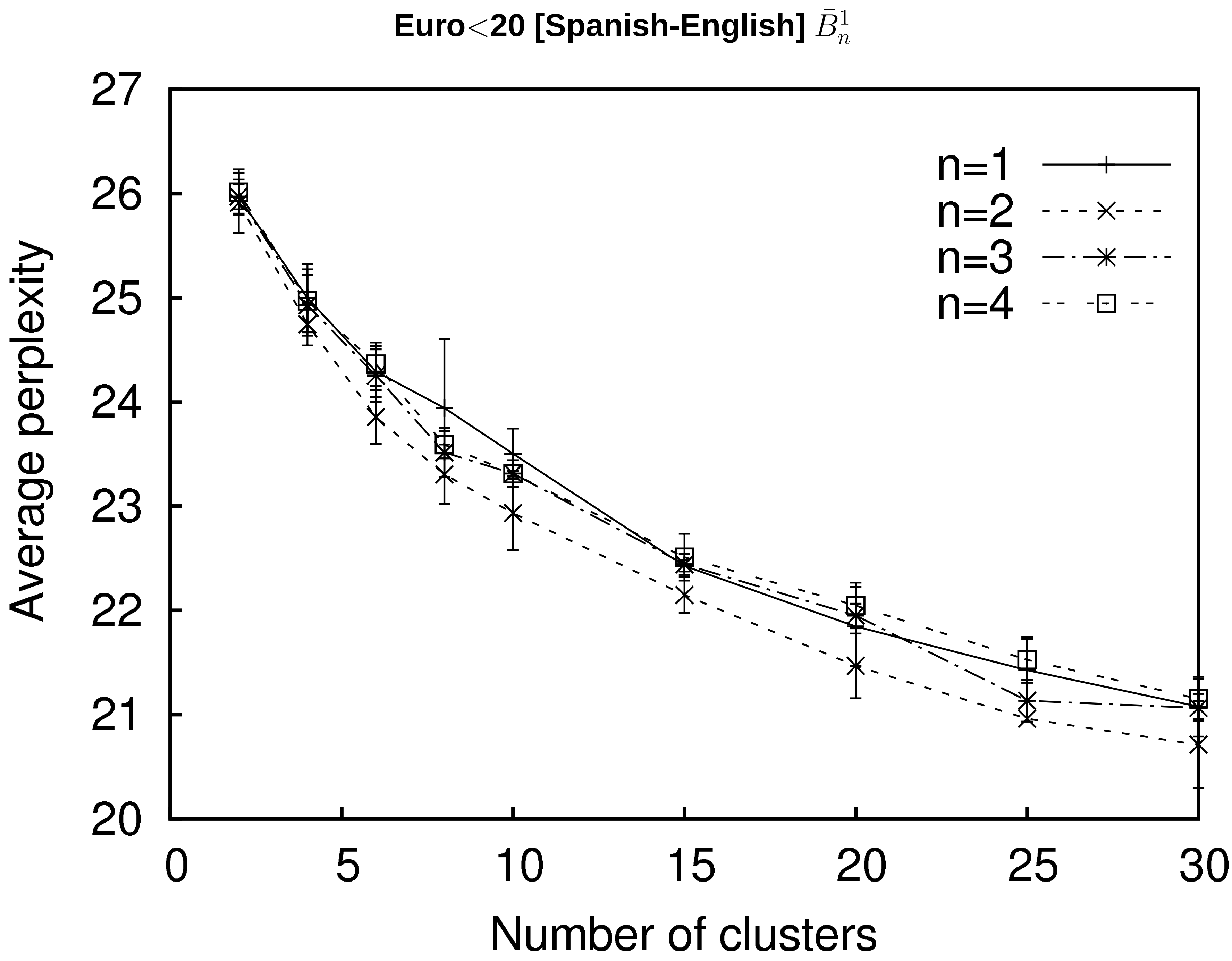


BTEC [Chinese-English] $\bar{K}_2^1$ and $\bar{B}_2^1$
17
16
15
14
13
12
11
10
2
4
6
8
10
12
14
16
18
20
Average perplexity
Number of clusters
random
monolingual
bilingual



BTEC [Chinese-English] $\bar{B}_n^1$
Average perplexity
Number of clusters
n=1
n=2
n=3
n=4
15
14
13
12
11
10
2
4
6
8
10
12
14
16
18
20


Euro<20 [Spanish-English] $\bar{B}_2^1$
Average perplexity
Number of clusters
random
monolingual
bilingual
29
28
27
26
25
24
23
22
21
20
0
5
10
15
20
25
30



Euro<20 [Spanish-English] $\bar{B}_n^1$
Average perplexity
Number of clusters
n=1
n=2
n=3
n=4
27
26
25
24
23
22
21
20
0
5
10
15
20
25
30

Why $n = 2$?

- 2-grams give more structural information than 1-grams

- But $3, 4$-grams give even more structural information

- Singletons and doubletons statistics

- Single stands for singletons and double for doubletons

- All figures are in %

	1-grams		2-grams		3-grams		4-grams	
Corpus	single	double	single	double	single	double	single	double
BTEC	43.8	14.0	65.3	13.6	79.0	10.5	87.5	7.5
Euro$<$20	36.7	13.3	62.7	13.3	78.9	9.8	88.4	6.2

- Almost all the $3, 4$-grams are not informative or little informative

6 Conclusions

- Kernels have been used as similarity measure in a clustering algorithm (C-means)
- Several families of kernels suitable for this task have been described
- The kernels $\bar{B}_2$ and $\bar{B}_2^1$ perform the best in practice
- No practical difference among $K_n^1(\ldots)$ and $K_n(\ldots)$ families
- In order to take advantage of bilingual information cluster sizes need to be large
- IC-PPL does not provide insight towards deciding the optimal number of clusters
- Which is the relationship between the distance and similarity clustering algorithms?
- Additional factors can be used in a bilingual-like extension
- Add stochastic indexing information by making $\mathbf{z}_n \in [0.0, 1.0]$

Thank you !