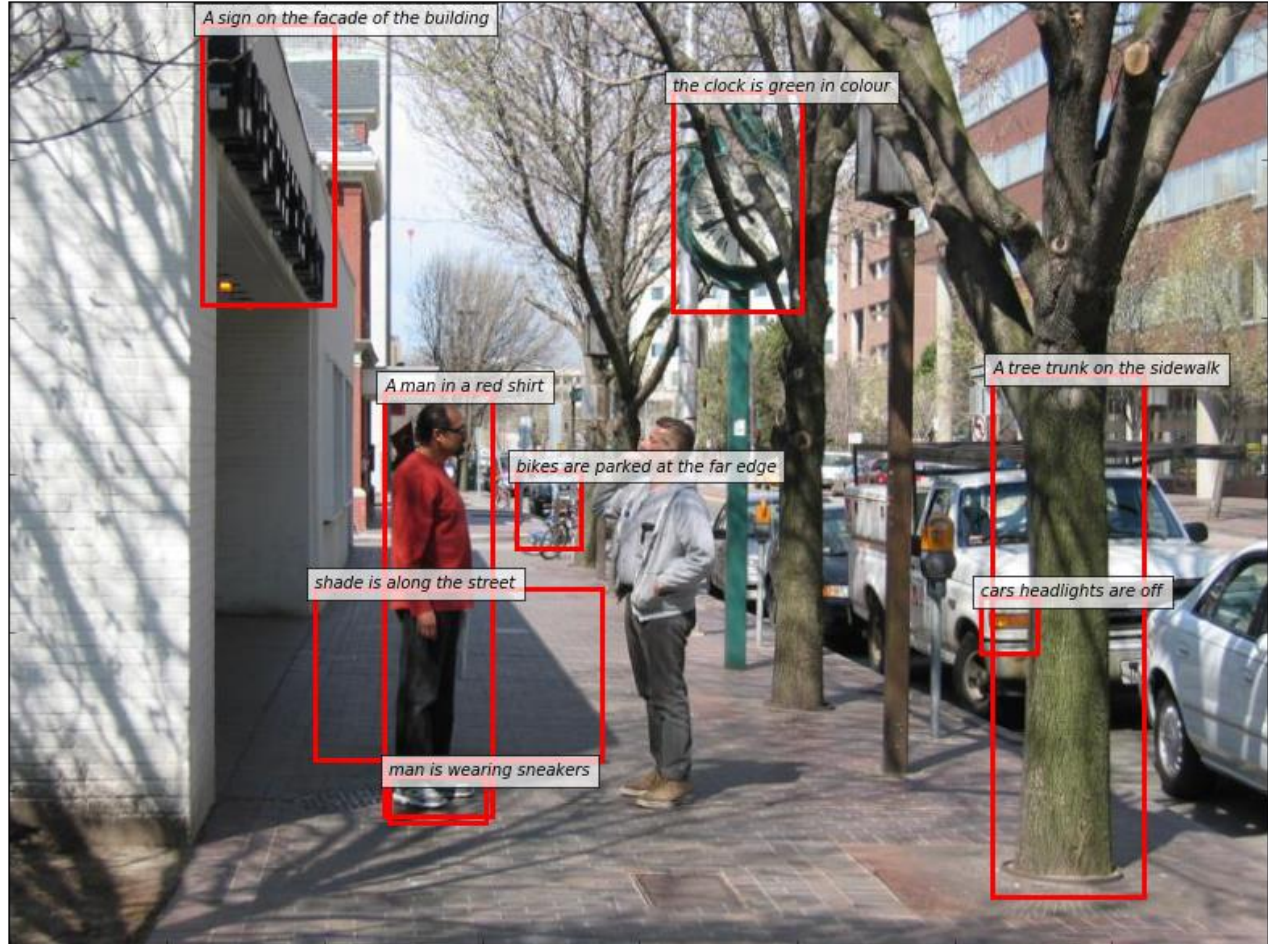




جامعة حلب  
كلية الهندسة المعلوماتية  
قسم الذكاء الصناعي  
السنة الرابعة



## كشف محتوى النصوص و الصور المترافقة في الويب الدلالي



### اشراف

د. فاضل سكر

### اعداد الطلاب

غيث شماع

نهاد نابلسي

علي ديب

جودي علي

آب 2022  
مشروع السنة الرابعة

*To My Mother , Father and Brother ...*

*To My Mentor and Father Figure , Dr Fadel Sukkar ...*

*To My Friend , Maamoun Dery ...*

*Ghaith Chamaa*

## الملخص

في هذا العمل تم إنشاء نظام يقوم بتوليد نصوص واصفة للمحتوى الموجود في مجموعة من النصوص التي تحتوي علاقات بين الكائنات باستخدام تقنيات من الذكاء الصناعي و معالجة اللغات الطبيعية , تم ذلك عن طريق تكوين قاعدة بيانات تحتوي على كل المعلومات اللازمة , و من ثم تكييف هذه البيانات لإزالة أي تشويش قد يؤدي سير العملية , والتعبير عن النصوص باستعمال تقانة عصبية , ومن ثم التوجه لإحداث شبكة عصبية قادرة على القيام بالمطلوب.

الهدف من هذا العمل هو استخلاص المحتوى الدفين عن طريق جمل قصيرة توضح العلاقات بين الكائنات الممثلة في الصورة المقابلة للجمل القصيرة . يتيح المشروع القدرة على وصف العام لما يحدث في صورة ما يمكن تطبيق المشروع في مجال الويب و صفحات الانترنت لتوليد توصيف لصور تشرحها ووضع الشرح تحت الصورة في حال كانت سرعات التحميل بطيئة أو قلة المساحة على القرص الصلب.

## الفهرس:

1	..... الفصل الأول: المقدمة
2	..... الفصل الثاني: احدث قاعدة معطيات للبيانات المستخدمة
2	..... 1.2 شرح عن البيانات المستخدمة
7	..... 2.2 سبب دمج البيانات المستخدمة و منهجية الدمج فيما بينها
12	..... الفصل الثالث: معالجة البيانات و تكييفها
13	..... 1.3 استخلاص أزواج التدريب اللازمة
13	..... 2.3 خط المعالجة الكلاسيكي لمعالجة البيانات
16	..... 3.3 الترميز باستعمال خوارزمية تشفير أزواج البايتات
17	..... 4.3 تكوين أشعة التضمين باستعمال خوارزمية Word2Vec
21	..... الفصل الرابع: بناء و تدريب النموذج
21	..... 2.4 لمحة عن خوارزمية الانتشار العكسي عبر الزمن
24	..... 2.4 لمحة عن وحدة الذاكرة طويلة قصيرة الأمد
25	..... 3.4 لمحة عن شبكة تشفير - فك تشفير
26	..... 4.4 التدريب بمنهجية إجبار المعلم
26	..... 5.4 مرحلة التدريب و الفحص
31	..... الفصل الخامس: الاستنتاجات و المقترحات
33	..... المراجع
	الملحق:
35	..... أ- ملحق أكواد الاستعلامات (Queries)

## المقدمة

الهدف من المشروع هو تحويل أي نص يحوي علاقات واصفة للكائنات المتواجدة في صورة ما إلى نص وصفي قادر على احتواء المعاني و المحتوى المعبر عنه من قبل الجمل الواصفة , وبوقت قصير نسبيا إلى الوقت الذي يستلزمه العمل المعتاد لإنتاج هذه الجملة واصفة خاصة في كانت الصورة غزيرة بالكائنات , وكل هذا تم باستخدام طرائق من الذكاء الصناعي ومعالجة اللغات الطبيعية لتحليل النص , و تطبيق شبكة عصبونية قادرة على تحقيق المراد الحصول عليه بأقل كلفة حسابية ممكنة.

وفيما يلي شرح موجز عن محتوى كل نص وأهميته:

**الفصل الثاني** احداث قاعدة معطيات للبيانات المستخدمة: يحتوي هذا الفصل على الأسس التي تركز عليها عملية إحداث قاعدة البيانات المستخدمة في المشروع .

**الفصل الثالث** معالجة البيانات و تكييفها للنموذج: يحتوي هذا الفصل على شرح المسار الكلاسيكي لمعالجة البيانات النصية كيف تم تمثيلها ، وعلى الطرق والعمليات التي تم استخدامها و شرحها.

**الفصل الرابع** بناء و تدريب النموذج: يحتوي هذا الفصل على شرح المنهجية المتبعة في بناء وتدريب الشبكة المستخدمة .

**الفصل والأخير** الاستنتاجات والمقترحات: يحتوي هذا الفصل على الاستنتاجات الأخيرة التي تخص هذا العمل والمقترحات لتطويره مستقبلا.

## احداث قاعدة معطيات للبيانات المستخدمة (Creating a Databased Dataset)

### 1.2 شرح عن البيانات المستخدمة (Demonstration of The Dataset Visual Genome):

تتألف البيانات من 15 ملف (13 ملف بصيغة JSON) و الملفان الآخران يحتويان على الصور بصيغة JPG , تم الحصول عليها من موقع الورقة البحثية [1] .  
نستعرض أسماء الملفات التي تم استخدامها و محتوياتها.

1. Attributes: يتألف من مصفوفة ب 108,076 عنصر و هذا مساوي لعدد الصور , كل عنصر يعبر عن مواصفات (attributes) لكل غرض من الأغراض (objects) في الصورة الواحدة , حيث يتألف العنصر الواحد , من المصفوفة السابقة , من هوية أو رقم الصورة (image\_id) و مصفوفة يحتوي كل عنصر منها مواصفات (attributes) غرض واحد من الأغراض (objects) الموجودة في الصورة الواحدة , كل عنصر منها يحوي على هوية الغرض (object\_id) و طول وعرض و احداثيات الزاوية العليا اليسرى للصندوق المحيط للغرض , و مصفوفة من المواصفات (attributes) للغرض و مصفوفة (names) تدل على الأسماء المختلفة للغرض و مصفوفة (synsets) التي تعني المفردات المطابقة لأسماء الخواص لهذا الغرض.

بنية الملف (schema):

```
{ '$schema': 'http://json-schema.org/schema#',  
  'type': 'array',  
  'items': { 'type': 'object',  
    'properties': { 'image_id': { 'type': 'integer' },  
      'attributes': { 'type': 'array',  
        'items': { 'type': 'object',  
          'properties': { 'synsets': { 'type': 'array', 'items': { 'type': 'string' } },  
            'h': { 'type': 'integer' },  
            'object_id': { 'type': 'integer' },  
            'names': { 'type': 'array', 'items': { 'type': 'string' } },  
            'w': { 'type': 'integer' },  
            'attributes': { 'type': 'array', 'items': { 'type': 'string' } },  
            'y': { 'type': 'integer' },  
            'x': { 'type': 'integer' } },  
          'required': [ 'h', 'names', 'object_id', 'synsets', 'w', 'x', 'y' ] } },  
        'required': [ 'attributes', 'image_id' ] } }
```

2. Attributes Synsets: خريطة أو قاموس (dictionary or map) مؤلفة من أزواج مفاتيح و قيم (key and

value pairs) توضح معاني المختلفة لكل خاصية.

لا يمكن عرض بنية الملف الاصلي بسبب هيكلته الضعيفة لذلك يجب القيام بإعادة هيكلته , حيث كانت النتيجة كالاتي:

### بنية الملف (schema):

```
{ '$schema': 'http://json-schema.org/schema#',  
  'type': 'array',  
  'items': { 'type': 'object',  
    'properties': { '_id': { 'type': 'string' },  
    'values': { 'type': 'array',  
      'items': { 'type': 'string' } },  
    'required': [ 'values', '_id' ] } }
```

3. Objects: يتألف من مصفوفة ب 108,076 عنصر و هذا مساوي لعدد الصور , كل عنصر يعبر عن الأغراض (Objects) في الصورة الواحدة , حيث يتألف العنصر الواحد , من المصفوفة السابقة , من هوية أو رقم الصورة (image\_id) و مصفوفة يحتوي كل عنصر منها على العناصر (Objects) الموجودة في الصورة الواحدة , كل عنصر منها يحوي على هوية الغرض (object\_id) و مصفوفة تدعى بالهويات الأغراض المدمجة (merged object ids) وتفيد بربط الغرض الواحد بعدة مواصفات (سنرى ذلك في قسم آلية الربط) و طول وعرض و احداثيات الزاوية العليا اليسرى للصندوق المحيط للغرض , و مصفوفة (synsets) التي تعني المفردات المطابقة لأسماء الخواص لهذا الغرض.

### بنية الملف (schema):

```
{ '$schema': 'http://json-schema.org/schema#',  
  'type': 'array',  
  'items': { 'type': 'object',  
    'properties': { 'image_id': { 'type': 'integer' },  
    'objects': { 'type': 'array',  
      'items': { 'type': 'object',  
        'properties': { 'synsets': { 'type': 'array', 'items': { 'type': 'string' } },  
        'h': { 'type': 'integer' },  
        'object_id': { 'type': 'integer' },  
        'merged_object_ids': { 'type': 'array', 'items': { 'type': 'integer' } },  
        'names': { 'type': 'array', 'items': { 'type': 'string' } },  
        'w': { 'type': 'integer' },  
        'y': { 'type': 'integer' },  
        'x': { 'type': 'integer' } },  
        'required': [ 'h', 'names', 'object_id', 'synsets', 'w', 'x', 'y' ] },  
        'image_url': { 'type': 'string' },  
        'merged_object_ids': { 'type': 'array' } },  
        'required': [ 'image_id', 'objects' ] } }
```

4. Objects Synsets: خريطة أو قاموس (dictionary or map) مؤلفة من أزواج مفاتيح و قيم (key and value pairs) توضح معاني المختلفة لكل غرض. لا يمكن عرض بنية الملف الاصيلي بسبب هيكلته الضعيفة لذلك يجب القيام بإعادة هيكلته , حيث كانت النتيجة كالاتي:

بنية الملف (schema):

```
{'$schema': 'http://json-schema.org/schema#',
'type': 'array',
'items': {'type': 'object',
'properties': {'_id': {'type': 'string'},
'values': {'type': 'array',
'items': {'type': 'string'}}},
'required': ['values', '_id']}
```

5. Relationships: يتألف من مصفوفة ب 108,076 عنصر و هذا مساوي لعدد الصور , كل عنصر يعبر عن العلاقات (Relationships) في الصورة الواحدة , حيث يتألف العنصر الواحد , من المصفوفة السابقة , من هوية أو رقم الصورة (image\_id) و مصفوفة يحتوي كل عنصر منها على العلاقات (Relationships) الموجودة في الصورة الواحدة , كل عنصر منها يحوي على هوية الغرض (relationship\_id) و كائن المفعول به (object) يحوي نفس محتويات الملف (Objects) و كائن الفاعل (subject) يحوي نفس محتويات الملف (Objects) كلا الفعل و الفاعل لا يحتويان على الهويات الأغراض المدمجة الموجودة في الملف (Object) , و المسند إلى (predicate) , الذي يعني الفعل , و مصفوفة (synsets) التي تعني المفردات المطابقة للفعل.

بنية الملف (schema):

```
{'$schema': 'http://json-schema.org/schema#',
'type': 'array',
'items': {'type': 'object',
'properties': {'relationships': {'type': 'array',
'items': {'type': 'object',
'properties': {'predicate': {'type': 'string'},
'object': {'type': 'object',
'properties': {'h': {'type': 'integer'},
'object_id': {'type': 'integer'},
'merged_object_ids': {'type': 'array', 'items': {'type': 'integer'}}},
'synsets': {'type': 'array', 'items': {'type': 'string'}},
'w': {'type': 'integer'},
'y': {'type': 'integer'},
'x': {'type': 'integer'},
'names': {'type': 'array', 'items': {'type': 'string'}},
'name': {'type': 'string'}},
'required': ['h', 'object_id', 'synsets', 'w', 'x', 'y'],
'relationship_id': {'type': 'integer'},
'synsets': {'type': 'array', 'items': {'type': 'string'}}},
'items': {'type': 'object',
'properties': {'_id': {'type': 'string'},
'values': {'type': 'array',
'items': {'type': 'string'}}},
'required': ['values', '_id']}
```



```

'subject': {'type': 'object',
'properties': {'name': {'type': 'string'},
'h': {'type': 'integer'},
'synsets': {'type': 'array', 'items': {'type': 'string'}}},
'object_id': {'type': 'integer'},
'w': {'type': 'integer'},
'y': {'type': 'integer'},
'x': {'type': 'integer'},
'merged_object_ids': {'type': 'array', 'items': {'type': 'integer'}},
'names': {'type': 'array', 'items': {'type': 'string'}}},
'required': ['h', 'object_id', 'synsets', 'w', 'x', 'y']},
'required': ['object',
'predicate',
'relationship_id',
'subject',
'synsets']},
'image_id': {'type': 'integer'}},
'required': ['image_id', 'relationships']}]

```

6. Relationships Synsets: خريطة أو قاموس (dictionary or map) مؤلفة من أزواج مفاتيح و قيم (key and value pairs) توضح معاني المختلفة لكل علاقة.  
لا يمكن عرض بنية الملف الاصلي بسبب هيكلته الضعيفة لذلك يجب القيام بإعادة هيكلته , حيث كانت النتيجة كالاتي:

بنية الملف (schema):

```

{'$schema': 'http://json-schema.org/schema#',
'type': 'array',
'items': {'type': 'object',
'properties': {'_id': {'type': 'string'},
'values': {'type': 'array',
'items': {'type': 'string'}}},
'required': ['values', '_id']}]

```

7. Region Graphs: يتألف من مصفوفة ب 108,076 عنصر و هذا مساوي لعدد الصور , كل عنصر يعبر عن المناطق (Regions) في الصورة الواحدة , حيث يتألف العنصر الواحد , من المصفوفة السابقة , من هوية أو رقم الصورة (image\_id) و مصفوفة يحتوي كل عنصر منها على المناطق (Regions) الموجودة في الصورة الواحدة , كل عنصر منها يحوي على هوية المنطقة (region\_id) و مصفوفة علاقات (relationships) يحوي كل عنصر منها على الفعل و هوية الفاعل و المفعول به و العلاقة , وطول و عرض و احداثيات الزاوية العليا اليسرى للصندوق المحيط للغرض و جملة (phrase) توضح المنطقة بشكل مقتضب , و مصفوفة الأغراض التي تحوي نفس محتويات الملف (Objects) ولا تحتوي على الهويات الأغراض المدمجة الموجودة في الملف (Object) , و مصفوفة (synsets) التي تعني المفردات المطابقة لأسماء الأغراض في المنطقة مضافا عليها مواقع تواجد الأغراض (objects) في الجملة.

بنية الملف (schema):

```

{'$schema': 'http://json-schema.org/schema#',
'type': 'array',
'items': {'type': 'object',
'properties': {'regions': {'type': 'array',
'items': {'type': 'object',
'properties': {'relationships': {'type': 'array',
'items': {'type': 'object',
'properties': {'synsets': {'type': 'array',
'items': {'type': 'string'}}},
'predicate': {'type': 'string'},
'relationship_id': {'type': 'integer'},
'object_id': {'type': 'integer'},
'subject_id': {'type': 'integer'}}},
'required': ['object_id',
'predicate',
'relationship_id',
'subject_id',
'synsets']}}},
'region_id': {'type': 'integer'},
'width': {'type': 'integer'},
'synsets': {'type': 'array',
'items': {'type': 'object',
'properties': {'entity_idx_start': {'type': 'integer'},
'entity_idx_end': {'type': 'integer'},
'entity_name': {'type': 'string'},
'synset_name': {'type': 'string'}}},
'required': ['entity_idx_end',
'entity_idx_start',
'entity_name',
'synset_name']}}},
'height': {'type': 'integer'},
'image_id': {'type': 'integer'},
'objects': {'type': 'array',
'items': {'type': 'object',
'properties': {'name': {'type': 'string'},
'h': {'type': 'integer'},
'object_id': {'type': 'integer'},
'synsets': {'type': 'array', 'items': {'type': 'string'}}},
'w': {'type': 'integer'},
'y': {'type': 'integer'},
'x': {'type': 'integer'}}},
'required': ['h', 'name', 'object_id', 'synsets', 'w', 'x', 'y']}},
'phrase': {'type': 'string'},
'y': {'type': 'integer'},
'x': {'type': 'integer'}},
'required': ['height', 'image_id', 'objects', 'phrase', 'region_id', 'relationships', 'synsets', 'width', 'x', 'y']},
'image_id': {'type': 'integer'}},
'required': ['image_id', 'regions']}]

```

8. Synsets: يتألف من مصفوفة ب 18,757 عنصر , كل عنصر يعبر عن الكلمة (synset\_id) , و التعريف المرافق لهذه الكلمة (synset\_definition).  
بنية الملف (schema):

```
{'$schema': 'http://json-schema.org/schema#',  
'type': 'array',  
'items': {'type': 'object',  
'properties': {'synset_definition': {'type': 'string'},  
'synset_name': {'type': 'string'}},  
'required': ['synset_definition', 'synset_name']}}
```

## 2.2 سبب دمج البيانات المستخدمة و منهجية الدمج فيما بينها

### :(The Reason Behind Merging The Dataset And The Methodology)

بإلقاء نظرة على الملفات السابقة و بنيتها (schema) , نجد أنه يوجد ترابط شديد بين هذه الملفات ولتحقيق الغاية المثلى من استخدام هذه البيانات , من الضروري أن يتم الدمج بينها (Normalizaion).  
المنهجية يمكن تلخيصها بالشجرة العلائقية (Relational Tree) {Fig 1} التالية :  
بما أن بنية الملفات تشعبية (Nested), فقد تم استخدام قاعدة بيانات غير علائقية (Non-Relational) وذلك لأن هذا النوع يدعم الوصول السريع للبيانات المتشعبة بسبب البنية الداخلية (Graph Database) على غرار النوع الآخر ألا وهو حفظ البيانات في جداول .

كمثال عن استعمال لجمع البيانات , المثال الآتي يستعرض استعمال (Query) للبيانات في المستوى السفلي في الشكل 1.1 (fig1.1) أي الأغراض (Objects):

```
// join between attributes and objects (images)
```

```
db.images.aggregate([  
  $unwind: "$objects"  
],  
{  
  $project: {  
    "objects.merged_object_ids": 1,  
    "objects.object_id": 1,  
    "image_id": 1  
  }  
},  
{  
  $lookup: {  
    from: "attributess",  
    let: {  
      localId: "$objects.object_id",  
      localImage: "$image_id"  
    },  
    pipeline: [  

```

```

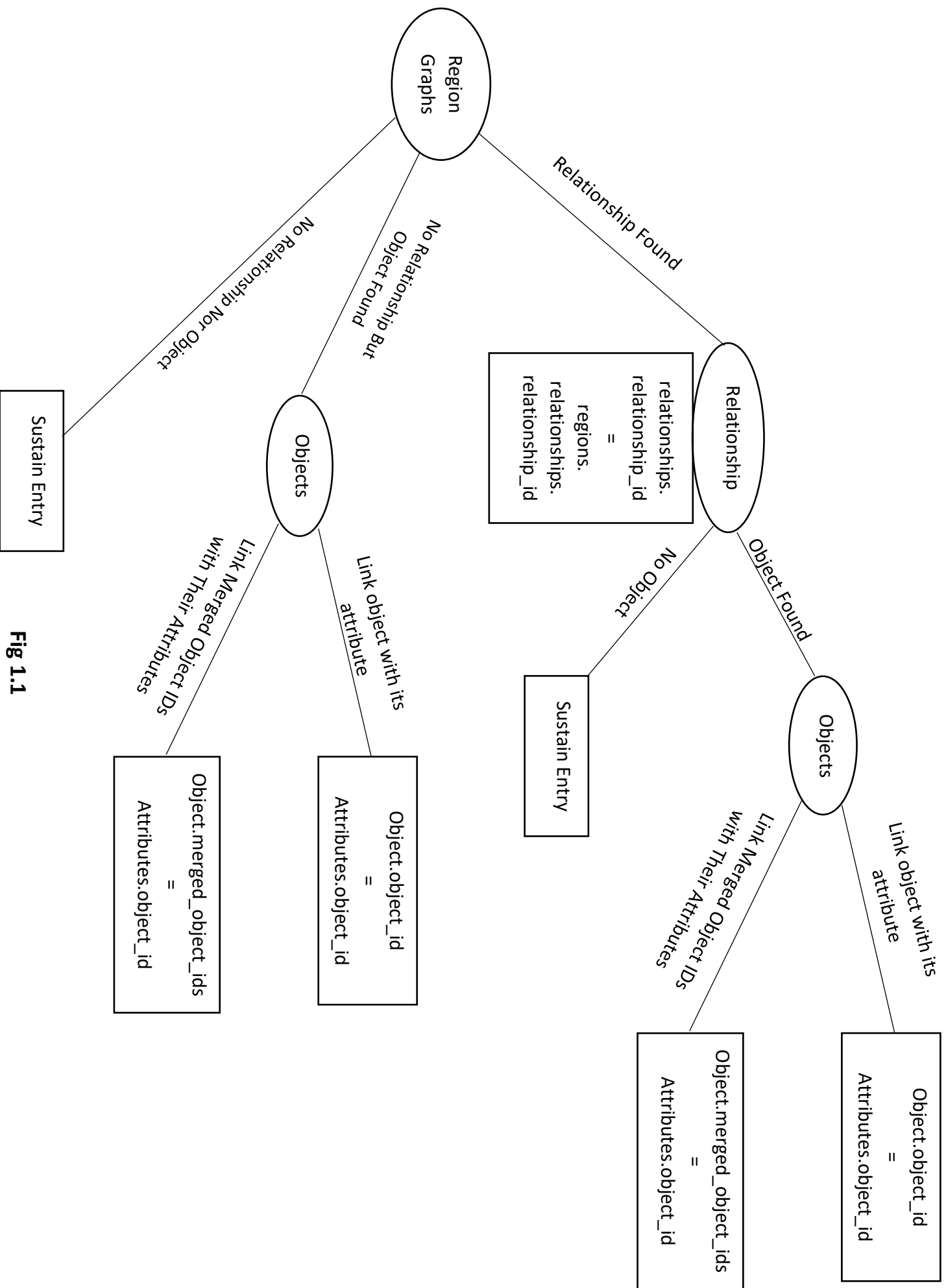
    { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
    { $unwind: "$attributes" },
    {
      $match: {
        $expr: {
          $eq: [
            "$attributes.object_id",
            "$$localId"
          ]
        }
      }
    },
    as: "x"
  }
},
{
  $lookup: {
    from: "attributess",
    let: {
      localId: "$objects.merged_object_ids",
      localimage: "$image_id"
    },
    pipeline: [
      { $match: { $expr: { $eq: ["$image_id", "$$localimage"] } } },
      { $unwind: "$attributes" },
      {
        $match: {
          $expr: {
            $in: [
              "$attributes.object_id",
              "$$localId"
            ]
          }
        }
      }
    ],
    as: "y"
  }
},
{
  $project: {
    "x": 1,
    result: {
      $reduce: {
        input: "$y.attributes.attributes",
        initialValue: [],
        in: { $concatArrays: ["$$value", "$$this"] }
      }
    }
  }
}

```

```

    }
  },
  {
    $project: {
      attributes: {
        $reduce: {
          input: "$x.attributes.attributes",
          initialValue: "$result",
          in: { $concatArrays: ["$$value", "$$this"] }
        }
      }
    }
  }
}]])

```



**Fig 1.1**  
10

عملية unwind تقوم بتحويل مصفوفة الأغراض (Objects) إلى كائنات مستقلة وذلك لتسهيل الترابط بين كل غرض (Object) و خصائصه.

عملية project تقوم بإسقاط (drop) جميع البيانات غير المراد الاحتفاظ بها و السماح للعناصر ذات العلم 1 بالبقاء في الذاكرة.

عملية lookup و هي العملية الأساسية في الترابط , تتشبه عملية Select في قواعد البيانات العلائقية , تقوم هذه العملية في المثال بالترابط بما يشابه Outer Join حيث يتم تحديد الملف attributes و الخواص التي سيتم أخذها كمفاتيح ربط ثم يتم بناء pipeline يسمح بتنفيذ عدة عمليات في مكان واحد , واحدة تلو الأخرى , أولها match بين هوية الصورة في الملف Objects (localImage) و المقابل لها في الملف attributes (image\_id) , عملية unwind مشابهة للمذكورة سابقا , وعملية match بين كل هوية غرض و المماثل لها في الملف attributes .

يتم تكرار نفس عملية lookup و لكن هذه المرة مع هويات ال merged\_object\_ids التي من فوائدها اكساب الغرض تعددية الخواص.

السبب الرئيسي في استحداث قاعدة البيانات هذه هو أننا نريد جعل كل البيانات المترابطة فيما بينها أن تكون في مكان واحد خاص لهذه الصورة فقط مما يسهل استخلاص اي بيانات قد نحتاج اليها في اثناء بناء النموذج model .

## الفصل الثالث

### معالجة البيانات وتكييفها للنموذج



## معالجة البيانات و تكيفها (Data Processing and Conditioning)

### 1.3 استخلاص أزواج التدريب اللازمة (Extraction of The Required Training Pairs):

تم صياغة أزواج التدريب على النحو التالي , حيث كل عينة دخل تكون تابعة لكل منطقة (region) في كل صورة (image):  
الدخل يكون عبارة عن الثلاثية : subject – predicate – object  
الخرج يكون عبارة عن الجملة المعبرة عن المنطقة المعنية : phrase

مثال : الدخل : " cow in grass "

الخرج : " cow's head in the grass "

ولكن كما رأينا في الفصل السابق يمكن أن يكون هناك أكثر من علاقة واحدة في المنطقة الواحدة وأي أكثر من ثلاثية لكل منطقة وسيتم معالجة ذلك بإدخال كلمة (رمز) جديد , سيتم توضيح ذلك في القسم 3.3 , ولكن كمثال عن الحالة السابقة نستعرض المثال الآتي :

مثال : الدخل : " cow in field " , " cow in field " , " cow in field "

الخرج : " three cows in a field "

### 2.3 خط المعالجة الكلاسيكي لمعالجة البيانات النصية

#### :(The Classical Pipeline for Processing Text Data)

من الضروري أن يتم تهيئة البيانات مهما كانت طبيعتها لكي تكون قابلة للفهم من قبل الشبكة العصبونية , و في مجال معالجة اللغات الطبيعية (NLP) يوجد مسار متفق عليه للقيام بذلك:

#### 1.2.3 إزالة كلمات التوقف (Stop Words Removal):

من أول و أبسط مراحل المعالجة حيث يتم حذف جميع الكلمات الموجودة في لائحة كلمات التوقف (Stop Word List) والتي تعتبر من أكثر الكلمات المستخدمة في لغة معينة ومنها احرف الجر (a , an , and ...etc) و ممكن أن تكون أفعال الكون أو الأفعال الأكثر استعمالا (am , is , are , do , does ...etc) .  
ولكن هذه الخطوة تم الاستغناء عنها لأنه في حال تطبيقها سيتم تشويه أغلب الثلاثيات حيث سيتم حذف المسند (Predicate) منها و ذلك يؤدي إلى فقد في البيانات

#### 2.2.3 إزالة الترقيم (Punctuation Removal):

يتعبر الترقيم في معالجة اللغات الطبيعية بمثابة التشويش (Noise) في إشارة الدخل , لذلك من الأفضل حذف الترقيم في حال التواجد .

تم تطبيق هذه الخطوة باستثناء إشارة الفاصلة العليا ( ' apostrophe ) لأنها تستخدم كوسيلة للدمج بين الفاعل و المسند أو الفعل (Subject and Predicate) , كمثال : he's ⇒ he + is

#### 3.2.3 الاشتقاق و التجزيع (Stemming and Lemmatization):

لأسباب نحوية , تستخدم النصوص أشكالاً مختلفة من الكلمات , مثل: organize, organizes, and organizing بالإضافة إلى ذلك , هناك عائلات من الكلمات ذات الصلة الاشتقاقية لها معاني متشابهة , مثل: democracy,

democratic, and democratization الهدف من كل من الاشتقاق التجزيع هو تقليل أشكال التصريف وأحيانًا الأشكال الاشتقاقية ذات الصلة الواحدة للكلمة إلى صيغة قاعدية مشتركة. على سبيل المثال: am, are, is ⇒ be . لم يتم تطبيق هذه العملية لأنها ستقوم بالقضاء على التنوع الصرفي والدلالي للكلمات .

### 4.2.3 الترميز (Tokenization):

هذه العملية تقوم بتحويل أزواج التدريب إلى وحدات اصغر منها تدعى رمز (Token) , هذه الرموز يتم ربطها بأرقام ليتم اعطاءها للنموذج أو يتم استعمالها في خطوة لاحقة قبل الادخال إلى النموذج , الحاجة لهذه العملية تأتي من طرح السؤال الآتي : " كيف يمكننا أن نجعل الحاسوب قادر على القراءة البيانات النصية ؟ " , للوهلة الأولى يمكن أن يتم تعريف مجموعة من القواعد لتعريف قاموس و لكن هذا لا يفيد لأننا نحتاج لأن نعلم النموذج هذه القواعد . يوجد العديد من الطرق للترميز , نستعرض بعضها من الأبسط حتى الأبعد:

#### 1.4.2.3 الترميز حسب الكلمة (Word Based Tokenization):

من الطرق الأبسط و الأسرع حيث تعتمد على تجزئة النص حسب الكلمة أي حسب الفراغ (Space) بين كل كلمة . ولكن يوجد خطورة باحتمال كبير أن يوجد كلمات مفقودة في فترة فحص النموذج (Testing) بسبب التحويلات (Variants) الممكنة لكل كلمة في لغة ما

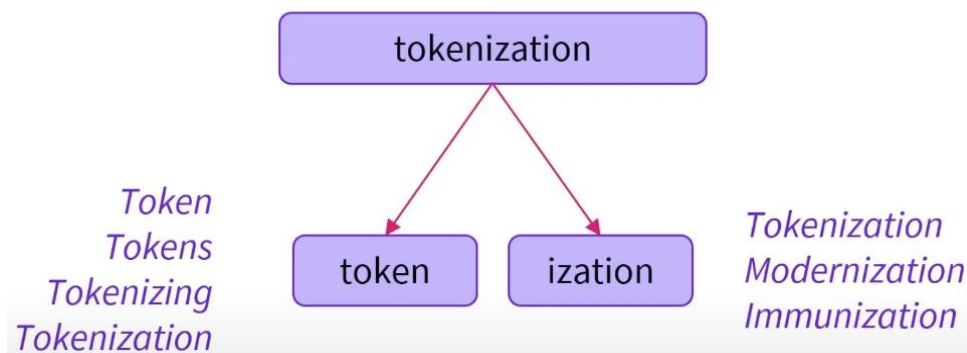
#### 1.4.2.3 الترميز حسب الحرف (Character Based Tokenization):

هذه الطريقة جاءت لحل المشكلة المتعلقة بالطريقة السابقة , ولكن هذه الطريقة ذاتها جاءت بمشاكل خاصة بها , أولها هو الحاجة الكبيرة للموارد الحاسوبية لمعالجة كل حرف بحد ذاته في النص كاملا , ثانيها هو تضيق أفق المجالات التطبيقية لأنه عند التعامل مع سلاسل طويلة من الأحرف هذا يقوم بإجبار استخدام شبكات عصبونية معينة , أخيرا يوجد خطورة فقدان المعاني (Semantics) أو حتى تعلمها بشكل خاطئ.

#### 1.4.2.3 الترميز حسب الكلمة الفرعية (Subword Tokenization):

في الطريقة الأولى نخاطر بعدم الإلمام بجميع التحويلات الممكنة لكلمة ما , وفي الطريقة الثانية نخاطر فقد المعاني , هنا تأتي الحاجة للطريقة الثالثة التي من أهدافها الاحتفاظ بالمعاني مع القدرة على الإلمام بمفردات كثيرة , وللوصول لهذه الأهداف يمكن أن نفكر بالكلمة بأنها تتألف من سابقة (Prefix) و لاحقة (Suffix) .  
كمثال: unhappily ⇒ un + happ + ily

ولكن هذه الطريقة ليست بالحل الأمثل لأنه يمكن أن يكون هناك بعض الرموز المنشئة لا يتم استخدامها أبدا للترميز عند الدخول لهذه المرحلة مما يؤدي إلى أخذ ذاكرة دون جدوى , للتخفيف من آثار هذه المشكلة تم يوجد مرمزات (Tokenizers) التي يتم تدريبها قبل الترميز , يوجد أيضا مشكلة مهمة أو جانب لاستخدام هذه التقنية و هو أن يوجد معامل (Hyperparameter) يحدد حجم الرموز المخرجة من المرمز , فإذا كان كبيرا سيقترب المرمز الى ترميز حسب الكلمة أما إذا كان صغيرا فسيقترب المرمز الى ترميز حسب الحرف , يوجد الكثير من الخوارزميات منها خوارزمية تشفير أزواج البايتات (Byte Pair Encoding) التي سيتم شرحها في القسم 3.3.



### 5.2.3 حساب الدخول باستخدام أشعة التضمين (Calculating Embedding Vectors):

إن خوارزميات التضمين تقوم بربط الرمز بأشعة ذات قيم حقيقية ، هذا يتم عن طريق ترميز كل كلمة في كل سلسلة و ترميمها و من ثم تضمينها (Embed) في فضاء شعاعي (Vector Space) . أشعة التضمين تهدف إلى التقاط المعنى الدلالي للكلمات في تسلسل نصي ، حيث تقوم الخوارزمية بتعيين تمثيلات عددية مماثلة (Numerical Representation) للكلمات ذات المعاني المشابهة. يوجد العديد من الطرق لحساب أشعة التضمين نستعرض بعضها من الأبسط حتى الأبعد:

#### 1.5.2.3 TF-IDF — Term Frequency-Inverse Document Frequency

تعتمد هذه الخوارزمية على مقياس إحصائي للعثور على لإيجاد صلة الكلمات في النص ، يمكن أن يكون النص في شكل مستند أو مستندات متنوعة (Corpus) ، إنه مزيج من مقياسين: تردد الكلمة (TF) وتردد المستند العكسي (IDF). تعتمد TF على تكرار الكلمات في المستند. يتم حساب الكلمات كعدد تكراراتها في المستندات ، يتم حساب TF بقسمة عدد تكرارات الكلمة (i) على العدد الإجمالي (N) للكلمات في المستند (j):

$$TF(i) = \log(\text{frequency}(i,j)) / \log(N(j))$$

IDF تحسب ندرة الكلمات ، إنه مهم لأن TF تعطي وزناً أكبر للكلمات التي تحدث بشكل متكرر. ومع ذلك ، فإن الكلمات التي نادراً ما تستخدم في المحتوى (Corpus) قد تحتوي على معلومات مهمة ، يمكن حسابه بقسمة العدد الإجمالي (N) للمستندات (d) على عدد المستندات التي تحتوي على الكلمة (i):

$$IDF(i) = \log(N(d) / \text{frequency}(d,i))$$

يتم أخذ  $\log$  في الصيغ أعلاه لتخفيف تأثير القيم الكبيرة لـ TF و IDF ، يتم حساب النتيجة النهائية لـ TF-IDF بضرب درجات TF و IDF. تُستخدم خوارزمية TF-IDF في حل المشكلات البسيطة ، يتم استخدامه بشكل أفضل لاسترجاع المعلومات (information retrieval) ، واستخراج الكلمات الرئيسية (keyword extraction) ، وإزالة كلمات التوقف. لا يمكنه التقاط المعنى الدلالي للكلمات في تسلسل بكفاءة.

#### 2.5.2.3 Word2Vec — Capturing Semantic Information

طورت من قبل Mikolov et al [3,4] Word2Vec هي تقنية تضمين لحل مشاكل معالجة اللغات الطبيعية المتقدمة ، يمكنه العبور على مجموعة كبيرة من النصوص (Corpus) لتعلم الارتباطات (associations) أو التبعية (dependencies) بين الكلمات ، يجد Word2Vec التشابه (Similarity) بين الكلمات باستخدام مقياس تشابه جيب التمام (Cosine Similarity) إذا كانت زاوية جيب التمام 1 ، فهذا يعني أن الكلمات متداخلة ، أما إذا كانت زاوية جيب التمام 90 ، فهذا يعني أن الكلمات مستقلة أو لا تحتوي على تشابه سياقي (Contextual). تستعمل Word2Vec إحدى الشبكتين العصبيتين: حقبة الكلمات المستمرة (Continuous Bag of Words (CBOW) و Skip-gram . الشكل 1.2 (fig 1.2) يوضح بنيتهما .

في CBOW ، يأخذ نموذج الشبكة العصبية كلمات مختلفة كمدخلات ويتنبأ بالكلمة المستهدفة التي ترتبط ارتباطاً وثيقاً بسياق كلمات الإدخال. من ناحية أخرى ، تأخذ بنية Skip-gram كلمة واحدة كمدخل وتتوقع كلمات السياق وثيقة الصلة بها. CBOW سريع و يجد تمثيلات رقمية أفضل للكلمات المتكررة ، بينما يمكن لـ Skip Gram تمثيل الكلمات النادرة بكفاءة. تعد نماذج Word2Vec جيدة في التقاط العلاقات الدلالية بين الكلمات.

Word2Vec هو الأنسب لإجراء التحليل الدلالي (Semantic Analysis) واكتشاف المعرفة (knowledge discovery).

سيتم التعمق في هذه الخوارزمية في القسم 4.3.

### 3.5.2.3: GloVe — Global Vectors for Word Representation

طورت من قبل Jeffery Pennington et al , يوسع GloVe عمل Word2Vec لالتقاط المعلومات السياقية العامة في مجموعة نصية عن طريق حساب مصفوفة التواجد المشترك العامة للكلمة (Global word-word co-occurrence Matrix) , يلتقط Word2Vec السياق المحلي للكلمات فقط. أثناء التدريب ، يأخذ بعين الاعتبار الكلمات المجاورة فقط لالتقاط السياق. يأخذ GloVe في الاعتبار المجموعة (Corpus) بأكملها وينشئ مصفوفة كبيرة يمكنها التقاط التكرار المشترك للكلمات داخل المجموعة (Corpus). يعمل GloVe بشكل أفضل في تشبيه الكلمات والتعرف على الكيانات المسماة (Named Entity Recognition). إنه أفضل من Word2Vec في بعض المهام وينافس في أخرى. ومع ذلك ، فإن كلا الأسلوبين جيدان في التقاط المعلومات الدلالية (Semantic Information) داخل مجموعة (Corpus).

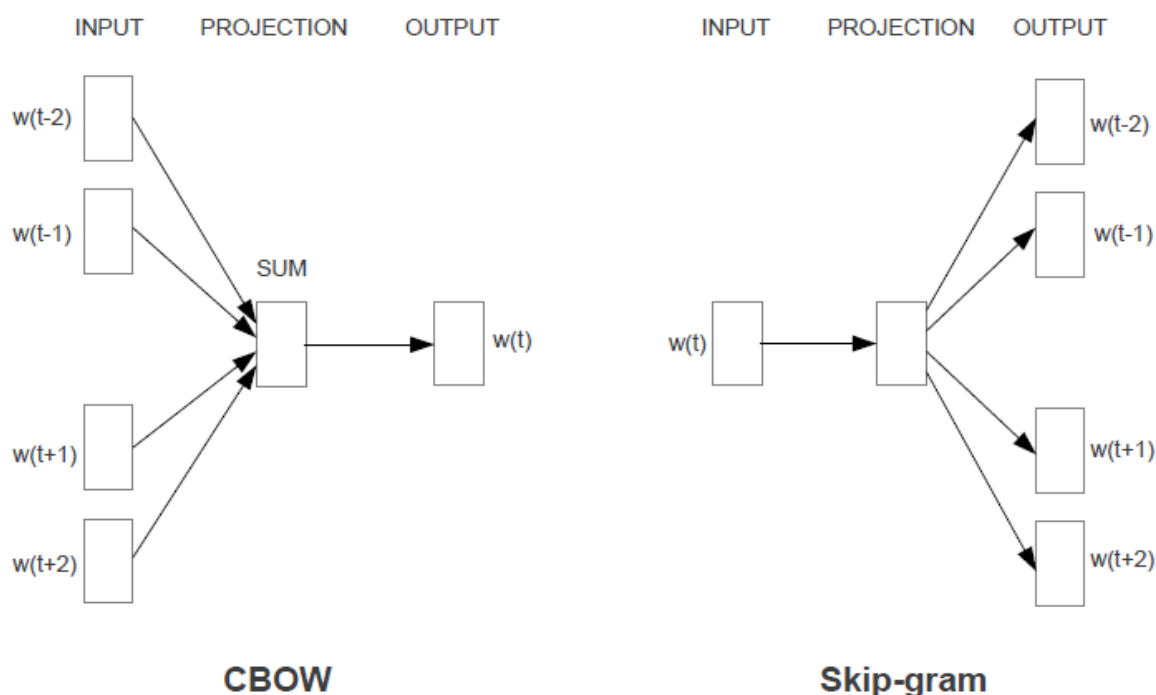


Fig 1.2

### 3.3 الترميز باستعمال خوارزمية تشفير أزواج البايتات (Tokenization Using Byte Pair Encoding)

إن خوارزمية BPE تستخدم لضغط البيانات حيث أن زوج البايتات الأكثر تكرارا يستعاض ببايت لا يوجد في البيانات الأصلية.

ليكن لدينا السلسلة aaabdaaabc التي تحتاج لضغط (تشفير) ان البايت الأكثر تكرارا هو aa يتم الاستعاضة عنه ب Z , ثاني أكثر زوج بايت مكرر هو ab يتم التبديل مع Y , بقي آخر زوج بايت و هو ac وبما أنه تكرر مرة واحدة لا يتم تشفيره , يمكن أن نقوم بتشفير تعاودي , عندها سيتم تشفير ZY ب X , تصبح السلسلة النهائية بالشكل XdXac حيث :  $X = ZY$  ,  $Y = ab$  ,  $Z = aa$  , لا يمكن التشفير أكثر لأنه لا يوجد أزواج بايت مكررة أكثر من مرة.

تم اقتراح تكييف هذه الخوارزمية و تطبيقها لمعالجة اللغات الطبيعية من قبل Sennrich et al [2] , حيث بداية يتم تهيئة كل كلمة على شكل محرفي (كل حرف على حدا) الكلمة أعلاه تم تهيئتها لتصبح : ('a','a','b','d','a','a','a','b','a','c') مع إضافة محرف جديد </w> الذي يعبر عن المسافة (White Space) لكي نكون قادرين على فك التشفير.

إن فرضية [1] هي أن تقسيم الكلمات النادرة الى كلمات فرعية مناسبة هو كافي لجعل الشبكة العصبونية قادرة تعميم المعرفة و خلق كلمات لم يتم رؤيتها من قبل من قبل الشبكة.

إن معيار عدد مرات الدمج هو المعامل الوحيد (Hyper Parameter) , الفارق بين هذه الخوارزمية و أي خوارزمية تشفير أو ضغط كخوارزمية Huffman حسب [2] , هو أن تسلسلات المحارف يمكن التعرف عليها ككلمة فرعية (Subword) وأن الشبكة العصبونية يمكنها أن تعمم لتكوين كلمات جديدة (لم يتم ورودها في التدريب) بالاعتماد على هذه الكلمات الفرعية , في مرحلة الفحص يتم تجزئ الكلمات الى محارف و تطبيق عمليات الدمج التي تم تعلمها في فترة التدريب , لكي يتم تشكيل كلمات معروفة .

تم تطبيق منهجيتان للاختبار في [1] وهما تطبيق BPE على عينة التدريب المدخلة مستقلة عن تطبيق الخوارزمية على عينات التدريب المخرجة أو تطبيق BPE على كليهما معا , ما يسمى Joint BPE , الطريقة الأولى لها أفضلية ضغط أعلى و تضمن بأن كل كلمة فرعية قد تم رؤيتها في مجالها الخاص (الدخل مستقل عن الخرج) , أما الطريقة الثانية فتحسن من التناسق بين الدخل و الخرج , ولكن وجد أيضا أن بتطبيق الطريقة الأولى (الطريقة المستقلة) يوجد احتمال كبير في تكوين كلمات فرعية مختلفة لنفس الكلمة مما يؤدي إلى جعل عملية تعلم الشبكة على خلق ارتباطات بين الكلمات الفرعية صعبا جدا.

تم تبني المنهجية الثانية أي Joint BPE في تشفير بيانات المشروع , حيث تم وضع المعامل حجم القاموس مساويا 400 مما يجعل معامل الدمج محسوبا تلقائيا , تم إضافة ستة رموز جديدة ليتم أخذها في الحسبان أثناء تنفيذ الخوارزمية , وهي:

- **<PAD>** : هذا الرمز تم استعماله لجعل جميع سلاسل الدخل و الخرج لها طول واحد وهو طول أطول سلسلة في كل من الدخل و الخرج , حيث يتم إضافة هذا الرمز بعد نهاية السلسلة.
- **<UNK>** : هذا الرمز يتم استعماله للتعامل مع الرموز غير الموجودة في قاموس الرموز (vocabulary)
- **<SEP>** : هذا الرمز يتم استعماله ليقيم بالتعبير عن وجود سلسلة أخرى بعد السلسلة الحالية , لأنه يمكن للدخل أن يكون عدة جمل أو سلاسل , لذلك يتم وضع هذا الرمز بين كل جملة و أخرى
- **<START>** : هذا الرمز يدل على بداية السلسلة , يتم وضعه لإعلام الشبكة ببداية السلسلة
- **<EMD>** : يشبه بوظيفته الرمز السابق . هذا الرمز يدل على نهاية السلسلة , يتم وضعه لإعلام الشبكة بنهاية السلسلة
- **</w>** : هذا الرمز تم ذكره سابقا , يتم وضعه بين كل كلمة و أخرى , يسمح بفك التشفير

### 4.3 تكوين أشعة التضمين باستعمال خوارزمية Word2Vec:

تم اختراع هذه الخوارزمية من قبل Mikolov et al [3,4] حيث تم اقتراح نموذجين أو شبكتين عصبونيتين هما Continuous Bag of Words (CBOW) و Skip Gram , لن يتم شرح CBOW لأنه تم تبني شبكة Skip Gram في هذا المشروع , لأن CBOW قوتها تكمن مع البيانات الضخمة و تمثيل أفضل للكلمات المتكررة [3] , فرضية [3] كانت بأن الكلمات الأكثر تشابها لها تمثيل رقمي متشابه و يمكن الحصول على درجات عميقة من التشابه , وهذا ما تم ملاحظته في اللغات الإعرابية أو التصريفية (Inflectional Languages) حيث أن المفردات يمكن أن يكون لها عدة نهايات و إذا تم البحث عن كلمات متشابهة في الفضاء الجزئي للفضاء الشعاعي الكلي , فإنه من الممكن وجود كلمات تنتهي بنهايات متشابهة.

تم اختيار Skip Gram في هذا المشروع لأنه وجد في [3,4] أنه يأخذ بالحسبان الكلمات النادرة , لأن بنيتها منطقية أكثر حيث أنه يتم توقع الكلمة المدخلة بناءً على الكلمات المجاورة للكلمة المدخلة و بسبب قدرتها الأعلى على كشف المعاني الدلالية من نظيرها CBOW.

إن تابع الكلفة [5] ل Skip Gram يهدف إلى الحصول على تمثيلات للكلمات  $c$  مفيدة لتوقع الكلمات المحيطة  $w$  للكلمة المدخلة في سلسلة ما  $D$  , للحصول على أشعة التضمين  $\theta$  العلاقة الرياضية هي :

$$\arg \max_{\theta} \prod_{(w,c) \in D} p(c|w; \theta)$$

تم اتباع منهجية الشبكات العصبونية و نمذجة العلاقة الاحتمالية الشرطية باستخدام تابع soft-max:

$$p(c|w; \theta) = \frac{e^{v_c \cdot v_w}}{\sum_{c' \in C} e^{v_{c'} \cdot v_w}}$$

حيث  $v_w$  و  $v_c$  يعبران عن التمثيل الشعاعي لل  $w$  و  $c$  , و يجب أن تكون البرامترات موضوعة بحيث يكون الجداء أعظمي. نأخذ log ليصبح الضرب جمع:

$$\arg \max_{\theta} \sum_{(w,c) \in D} \log p(c|w) = \sum_{(w,c) \in D} (\log e^{v_c \cdot v_w} - \log \sum_{c'} e^{v_{c'} \cdot v_w})$$

تم افتراض أن بتعظيم (Maximization) العلاقة السابقة سيولد أشعة تضمين جيدة بحيث الكلمات المتشابهة متقاربة رقميا. في [4,5] تم اقتراح تقنية أخذ العينات السلبية (Negative Sampling) , الذي يطرح السؤال التالي : هل الزوج  $(w,c)$  هو حقا من عينة التدريب ؟ , تابع الهدف الآن تغير , الهدف أصبح العثور على الأوزان  $\theta$  التي تعظم (Maximize) الاحتمال أن كل العينات جاءت من التدريب:

$$\begin{aligned} & \arg \max_{\theta} \prod_{(w,c) \in D} p(D = 1|w, c; \theta) \\ &= \arg \max_{\theta} \log \prod_{(w,c) \in D} p(D = 1|w, c; \theta) \\ &= \arg \max_{\theta} \sum_{(w,c) \in D} \log p(D = 1|w, c; \theta) \end{aligned}$$

يمكن أن نعرف المقدار  $p(D = 1|w, c; \theta)$  كالتالي:

$$p(D = 1|w, c; \theta) = \frac{1}{1 + e^{-v_c \cdot v_w}}$$

عندها يصبح تابع الهدف :

$$\arg \max_{\theta} \sum_{(w,c) \in D} \log \frac{1}{1 + e^{-v_c \cdot v_w}}$$

تابع الهدف يصبح عديم الأهمية (trivial) عندما وضع  $\theta$  بحيث  $p(D = 1|w, c; \theta)$  تساوي الواحد لكل زوج  $(w,c)$  , لذلك نحتاج الى آلية لمنع حدوث ذلك , عن طريق منع بضع من أزواج  $(w,c)$  من أن تكون ذو احتمال واحد أو أكيد , وذلك

عن طريق إضافة أزواج ليست موجودة في البيانات , يمكن ذلك عن طريق توليد مجموعة  $D'$  تحوي أزواج  $(w,c)$  بافتراض أن جميعها خاطئة (من هنا جاء اسم هذه التقنية):

$$\begin{aligned}
 & \arg \max_{\theta} \prod_{(w,c) \in D} p(D=1|c,w;\theta) \prod_{(w,c) \in D'} p(D=0|c,w;\theta) \\
 &= \arg \max_{\theta} \prod_{(w,c) \in D} p(D=1|c,w;\theta) \prod_{(w,c) \in D'} (1 - p(D=1|c,w;\theta)) \\
 &= \arg \max_{\theta} \sum_{(w,c) \in D} \log p(D=1|c,w;\theta) + \sum_{(w,c) \in D'} \log(1 - p(D=1|c,w;\theta)) \\
 &= \arg \max_{\theta} \sum_{(w,c) \in D} \log \frac{1}{1 + e^{-v_c \cdot v_w}} + \sum_{(w,c) \in D'} \log(1 - \frac{1}{1 + e^{-v_c \cdot v_w}}) \\
 &= \arg \max_{\theta} \sum_{(w,c) \in D} \log \frac{1}{1 + e^{-v_c \cdot v_w}} + \sum_{(w,c) \in D'} \log(\frac{1}{1 + e^{v_c \cdot v_w}})
 \end{aligned}$$

يسمح لنا أخذ العينات السلبية بتعديل نسبة صغيرة فقط من الأوزان ، بدلاً من جميع الأوزان لكل عينة تدريب , لذا فبدلاً من امتلاك soft-max عملاق واحد - التصنيف من بين 10000 فئة مثلاً ، تم تحويله الآن إلى 10000 مشكلة تصنيف ثنائي (sigmoid) .

تم وضع معامل حجم العينة الزائفة مساوياً 15 لأن حجم البيانات صغير نسبياً وذلك بناءً على [4] , تم تدريب الشبكة 50 مرة بناءً على [3] , تم جعل أشعة التضمين بطول 300 , الشكل 2.2 (Fig 2.2) يوضح بعض قيم التشابه (Similarity) بين بعض المرموز في أبجدية البيانات المستخدمة.

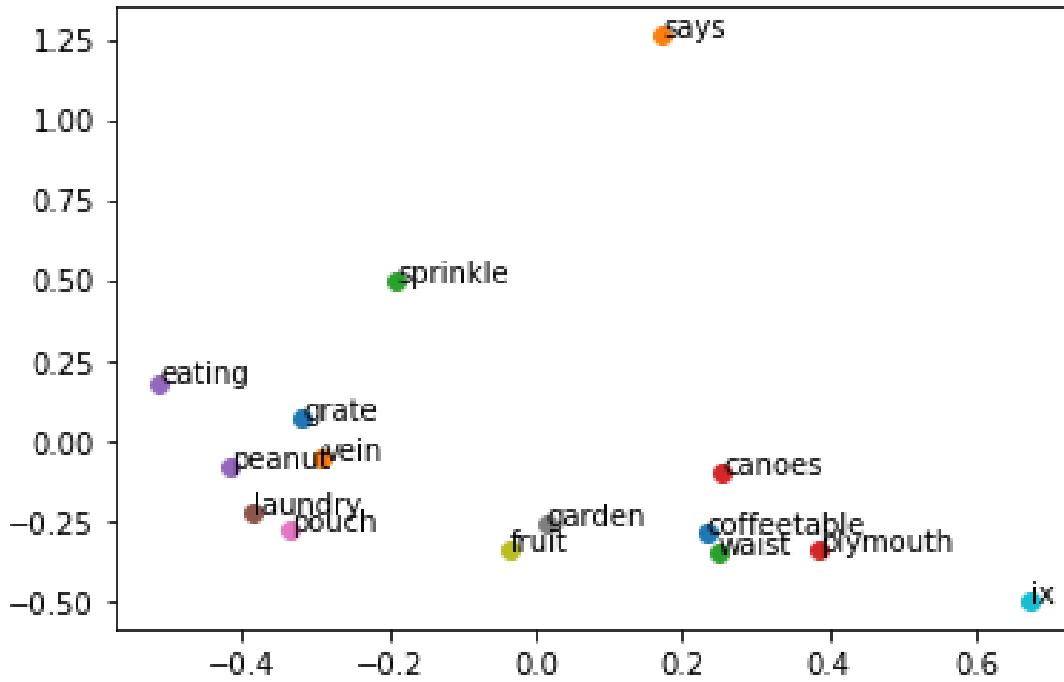


Fig 2.2

نجد تشابه منطقي بين الحديقة (garden) و الفاكهة (fruit) .

## الفصل الرابع

### بناء و تدريب النموذج



## بناء و تدريب النموذج (Model Building and Training)

### 1.4 لمحة عن خوارزمية الانتشار العكسي عبر الزمن (Back Propagation Through Time):

قبل أن يتم شرح خوارزمية الانتشار العكسي عبر الزمن , سيتم تغطية الأساسيات اللازمة لخوارزمية الانتشار العكسي التقليدية.

#### 1.1.4 أساسيات خوارزمية الانتشار العكسي التقليدية (Vanilla Back Propagation):

هذه الخوارزمية تفيد في حساب المشتقات بدقة و كفاءة لأي نظام ضخم مؤلف من أنظمة فرعية أساسية أو حسابات بحيث تكون ممثلة بتتابع رياضية معروفة لدى المستخدم قابلة للاشتقاق , لذا فإن لهذه الخوارزمية تطبيقات عديدة عدا الشبكات العصبونية.

كما نعلم , فإن مشكلة التعليم بإشراف (Supervised Learning) تهدف إلى تدريب شبكة عصبونية بخرج حقيقي  $\hat{Y}$  (Actual Output) لكي يكون أقرب ما يكون لخرج الهدف  $Y$  (Desired Output) لعينة تدريب تحوي  $T$  نموذج (Pattern) , الهدف هو تكييف أوزان الشبكة لكي تعمل بجودة لاستقبال أنماط خارج عينة التدريب. بالاستفادة من شبكة التدفق الأمامي البسيطة المذكورة في [6] , نعرفها كالتالي:

$$x_i = X_i, \quad 1 \leq i \leq m$$

$$net_i = \sum_{j=1}^{i-1} W_{ij} x_j, \quad m < i \leq N + n$$

$$s(z) = 1/(1 + e^{-z})$$

$$x_i = s(net_i), \quad m < i \leq N + n$$

$$Y_i = x_{i+N}, \quad 1 \leq i \leq n$$

حيث  $N$  هو عدد يمكن أن يكون ذو أي قيمة بشرط ألا يكون أقل من  $m$  , الشكل 1.4 fig يوضح البنية المذكورة في [6] :

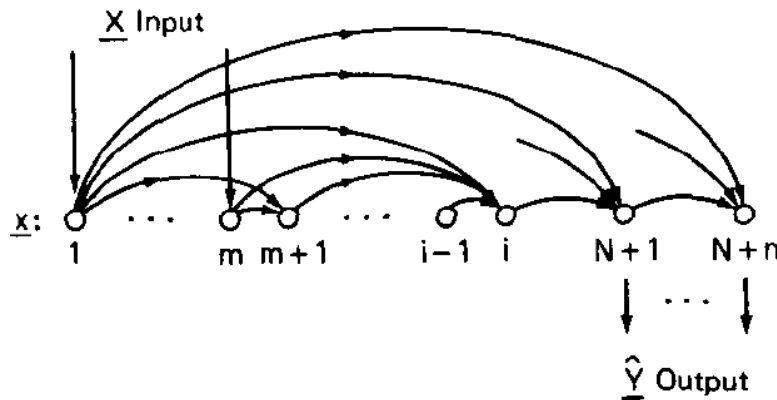


Fig 1.4

في علم الشبكات العصبية ، هذه الشبكة "متصلة بالكامل" (Fully Connected) ، من الناحية العملية ، في العادة يكون من المرغوب حد الاتصالات بين العصبونات ، يمكن القيام بذلك ببساطة عن طريق تصفير بعض الأوزان ليتم إسقاطها (Drop Out) من جميع العمليات الحسابية.

الهدف في الانتشار العكسي التقليدي هو تخفيض الخطأ التالي:

$$E = \sum_{t=1}^T E(t) = \sum_{t=1}^T \sum_{i=1}^n (1/2) (\hat{Y}_i(t) - Y_i(t))^2$$

الشكل 2.4 (fig 2.4) يوضح سير العمليات في الخوارزمية:

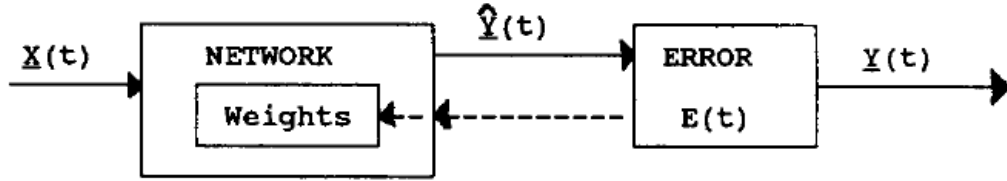


Fig 2.4

نحسب المخرجات  $Y(t)$  والأخطاء  $E(t)$  لتلك المجموعة الابتدائية من الأوزان. ثم نحسب مشتقات  $E$  بالنسبة إلى جميع الأوزان ، يشار إلى ذلك من خلال الخطوط المنقطة في الشكل أعلاه . إذا أدت زيادة وزن معين إلى مزيد من الخطأ ، فإننا نعدل ذلك الوزن لأسفل. إذا أدت زيادة الوزن إلى تقليل الخطأ ، نقوم بتعديله لأعلى. في [6] تم ابتكار صيغة جديدة لحساب الخطأ المتراكم كاشتقاق جزئي مرتب (Ordered Partial Derivatives) :

$$\frac{\partial^+ \text{TARGET}}{\partial z_i} = \frac{\partial \text{TARGET}}{\partial z_i} + \sum_{j>i} \frac{\partial^+ \text{TARGET}}{\partial z_j} * \frac{\partial z_j}{\partial z_i}$$

حيث تمثل المشتقات ذات الحرف المرتفع + المشتقات المرتبة والمشتقات بدون رموز تمثل المشتقات الجزئية العادية ، يمكن اعتبار كلا منهما على النحو التالي : تمثل المشتقات الجزئية البسيطة /التأثير المباشر لـ  $Z_i$  على  $Z_j$  ، أما المشتق المرتب يمثل التأثير الكلي لـ  $Z_i$  في الهدف (TARGET) .

توفر القاعدة المتسلسلة (Chain Rule) هذه طريقة مباشرة لكيفية حساب مشتقات متغير معطى (TARGET) فيما يتعلق بجميع المدخلات (و المعلمات) لنظام تفاضلي جزئي مرتب بمرور واحد فقط على النظام.

يمكننا كتابة المشتق المرتب لـ TARGET بالنسبة إلى  $z$  ، كـ "F-Z" الذي يمكن وصفه على أنه "التغذية الراجعة إلى  $Z_i$ ". في الانتشار العكسي التقليدي ، متغير الهدف هو الخطأ  $E$  ، بإخذ ما قيل بالحسبان ، تصبح العلاقة السابقة:

$$F_{-Z_i} = \frac{\partial E}{\partial z_i} + \sum_{j>i} F_{-Z_j} * \frac{\partial z_j}{\partial z_i}$$

والخطأ المرتكب في عصبونات الخرج هو :

$$F_{-\hat{Y}(t)} = \frac{\partial E}{\partial \hat{Y}_i(t)} = \hat{Y}_i(t) - Y_i(t)$$

باستخدام القاعدة التسلسلية في الاشتقاق لدينا:

$$F_{x_i}(t) = F_{\hat{Y}_{i-N}}(t) + \sum_{j=i+1}^{N+n} W_{ji} * F_{net_j}(t),$$

$$i = N + n, \dots, m + 1$$

$$F_{net_i}(t) = s'(net_i) * F_{x_i}(t), \quad i = N + n, \dots, m + 1$$

$$F_{W_{ij}} = \sum_{t=1}^T F_{net_i}(t) * x_j(t)$$

نلاحظ كيف يتطلب منا الرجوع للخلف عبر الشبكة من أجل حساب المشتقات .  
أخيرًا ، للتعديل الأوزان ، الطريقة المعتادة هي :

$$\text{New } W_{ij} = W_{ij} - \text{learning\_rate} * F_{W_{ij}}$$

الآن ، بعد أن تم تغطية خوارزمية الانتشار العكسي التقليدية ، سيتم الحديث عن خوارزمية الانتشار العكسي عبر الزمن.

#### 2.1.4 خوارزمية الانتشار العكسي عبر الزمن (Back Propagation through Time):

في بعض التطبيقات مثل التعرف على الكلام (Speech Recognition) ، سيكون التصنيف في الوقت  $t$  أكثر دقة إذا تمكنا من حساب ما رأيناه في أوقات سابقة ، إذا يجب استخدام فئة أكثر قوة من الشبكات للقيام بالتصنيف ، بحيث خرج الشبكة في الوقت  $t$  يأخذ في الحسابان المتغيرات في أوقات سابقة ، إن هذه الفئة تم الإطلاق على اسمها في [6] بشبكات التأخير الزمني أو التعاودية (Time Lagged or Recurrent) .

يمكن تطبيق Backpropagation على أي نظام له ترتيب محدد معرف بوضوح للحسابات ، حتى إذا كانت هذه الحسابات تعتمد على الحسابات السابقة داخل الشبكة نفسها ، بأخذ المثال المذكور في [6] الذي يتميز باعتماد زمني ذو رتبة ثانية ، يمكن حساب صافي (Net) كل عصبون بالعلاقة:

$$net_i(t) = \sum_{j=1}^{i-1} W_{ij} x_j(t) + \sum_{j=1}^{N+n} W'_{ij} x_j(t-1) + \sum_{j=1}^{N+n} W''_{ij} x_j(t-2).$$

عند تطبيق العلاقة السابقة عند  $t = 1$  ، سنواجه مشكلة تقنية بسيطة: لا توجد قيمة لـ  $x(-1)$  أو  $x(0)$  ، وكلاهما نحتاج إليه كمدخلات ، من حيث المبدأ ، يمكننا استخدام أي قيم نرغب في اختيارها ، يعد اختيار  $x(-1)$  و  $x(0)$  جزءًا أساسيًا من تعريف شبكتنا.

يمكن تكيف جميع المعادلات المذكورة في القسم السابق 1.1.4 لتسمح بوجود اعتماديات سابقة بالنسبة للزمن :

$$\begin{aligned} F_{x_i}(t) = & F_{\hat{Y}_{i-N}}(t) + \sum_{j=i+1}^{N+n} W_{ji} * F_{net_j}(t) \\ & + \sum_{j=m+1}^{N+n} W'_{ji} * F_{net_j}(t+1) \\ & + \sum_{j=m+1}^{N+n} W''_{ji} * F_{net_j}(t+2). \end{aligned}$$

لاحظ أن هذه المعادلة تجعل من المستحيل أن يتم حساب  $F_{net}(t)$  و  $F_{net}(t+1)$  إلا بعد معرفة كلا من  $F_{net}(t+1)$  و  $F_{net}(t+2)$  , لذلك لا يمكننا استخدام هذه المعادلة إلا من خلال العودة إلى الوراء في الوقت .

بالقاء نظرة على التوابع (Subroutines) في [6] , نجد أن هذه الخوارزمية تعمل عن طريق بسط (unroll) لكل المداخل حسب الزمن , كل لحظة زمنية هي نسخة من الشبكة بأوزانها الخاصة عند تلك اللحظة الزمنية ثم يتم حساب الأخطاء وتجميعها لكل خطوة زمنية. يتم لف (roll) الشبكة و تحديث الأوزان.

## 2.4 لمحة عن وحدة الذاكرة طويلة قصيرة الأمد

### :(Overview on The Long Term Short Memory Unit - LSTM)

تعلم تخزين المعلومات على فترات زمنية ممتدة عن طريق التكرار يستغرق Back Propagation وقتاً طويلاً جداً , ويرجع ذلك في الغالب إلى اضمحلال الخطأ الارتجاعي (Decaying Error Backflow) [7] , يمكن لـ LSTM تعلم جسر تأخيرات زمنية تزيد عن 1000 خطوة زمنية منفصلة عن طريق فرض تدفق خطأ ثابت من خلال دورات الخطأ الثابتة (Constant Error Carousel - CEC) داخل وحدات (بوابات) خاصة.

المشكلة كما ذكر وتم اثباته في [7] , أنه مع الانتشار العكسي التقليدي عبر الزمن , تميل إشارات الخطأ المتدفقة إلى الوراء في الوقت إلى (1) التفجير (Exploding) أو (2) الاختفاء (Vanishing) , يعتمد التطور الزمني للخطأ المنتشر بشكل أسّي على حجم الأوزان , وقد تؤدي الحالة 1 إلى تذبذب الأوزان , في الحالة الثانية , يستغرق تعلم جسر الفترات الطويلة وقتاً باهظاً أو لا يعمل على الإطلاق .

LSTM تم بناءها لحل المشكلة أعلاه , عن طريق خوارزمية فعالة قائمة على التدرج (Gradient-Based) لشبكة تفرض تدفق خطأ ثابت (وبالتالي , لا ينفجر ولا يتلاشى) من خلال الحالات الداخلية للوحدات الخاصة (البوابات - Gates).

إن الاثبات الرياضي لظاهرة تلاشي أو انفجار الخطأ موضح بشكل منطقي في [7] .

نستعرض الآن الوحدات الخاصة (البوابات) المذكورة في [7] التي حلت المشكلة , علماً أن الطريقة التي أدت إلى استنتاجها أيضاً مذكورة في [7] :

ليكن لدينا وحدة ذاكرة  $z$  (Memory cell) , يتم إضافة بوابة دخل ضاربة (Multiplicative Input Gate) لحماية محتويات الذاكرة للوحدة  $z$  من اضطراب بسبب مدخلات غير ذات الصلة. و يتم إضافة بوابة خرج ضاربة (Multiplicative Output Gate) لحماية الوحدات الأخرى من اضطراب بسبب محتويات الذاكرة الحالية المخزنة حالياً في  $z$  غير ذات الصلة.

الشكل 3.4 (fig 3.4) يوضح بنية وحدة الذاكرة  $z$  مع بواباتها ( $in\ z$  ,  $out\ z$ ) , يشير الاتصال الذاتي (مع الوزن 1.0) إلى تغذية راجعة بتأخير خطوة زمنية واحدة, هذا يشكل حجر الأساس للدورات (CEC) المذكورة في بداية القسم , وحدات البوابة تفتح وتغلق الوصول للدورات.

السبب وراء استخدام البوابات هو: لتجنب تعارض الأوزان مع المدخلات , تتحكم  $in\ z$  في تدفق الخطأ إلى اتصالات إدخال خلية الذاكرة  $z$  مع الأوزان الموافقة , بينما  $out\ z$  تتحكم في تدفق الخطأ إلى اتصالات مخارج خلية الذاكرة  $z$  مع الأوزان الموافقة .

بعبارة أخرى : يمكن للشبكة استخدام  $in\ z$  لتحديد اللحظة المناسبة للاحتفاظ بالمعلومات أو تجاوزها في خلية الذاكرة  $z$  و  $out\ z$  لتحديد اللحظة المناسبة للسماح للوحدات التي تلي  $z$  بالولوج لمعلومات  $z$  ومتى يتم منع الوحدات الأخرى من الاضطراب بواسطة  $z$  .

إذا يمكن أن نستنتج بأنه لا يمكن تغيير إشارات الخطأ المحتجزة داخل CEC لخلية الذاكرة ما .

قد يتم فرض إشارات خطأ مختلفة تتدفق إلى الخلية (في أوقات مختلفة) عبر بوابة الإخراج الخاصة بها، لنجد حينها أنه سيتعين على بوابة الإخراج معرفة الأخطاء التي يجب احتجازها في CEC من خلال معاييرها بشكل مناسب. وبشكل مماثل ، يجب أن تتعلم بوابة الإدخال متى يتم تحرير الأخطاء ، مرة أخرى عن طريق قياسها بشكل مناسب. بشكل أساسي ، تفتح وحدات البوابة الضاربة وتغلق الوصول إلى تدفق الخطأ المستمر عن طريق CEC.

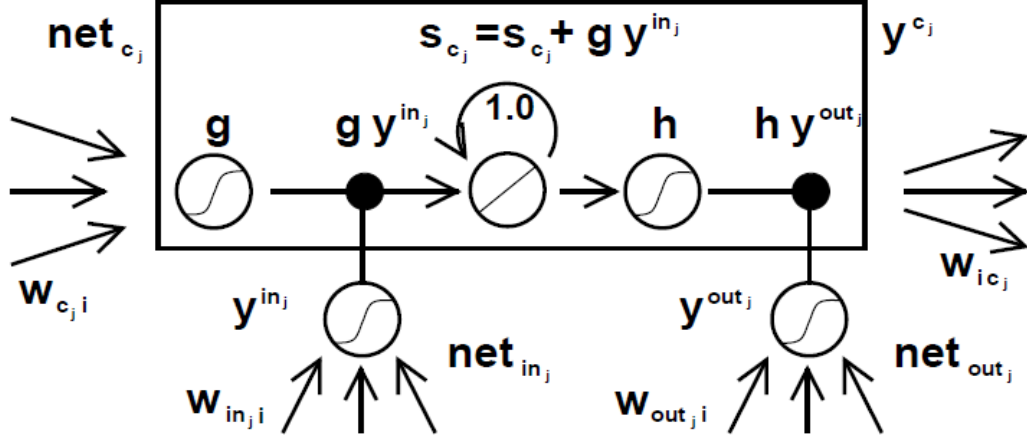


Fig 3.4

### 3.4 لمحة عن شبكة تشفير – فك تشفير (Overview on Seq2Seq Encoder-Decoder):

الشبكات العصبية العميقة (Deep Neural Networks - DNNs) هي نماذج قوية حققت نتائج ممتازة هي نماذج قوية حققت أداءً ممتازًا في مهام التعلم الصعبة. على الرغم من أن DNNs تعمل بشكل جيد كلما توفرت مجموعات تدريب كبيرة ، إلا أنه لا يمكن استخدامها لربط (Map) سلسلة من الدخل إلى سلسلة من الخرج الموافق للدخل (Sequence to sequence mapping)

تم في [8,9] اقتراح نموذج لشبكة عصبية حجرها الأساس هو وحدة LSTM المشروحة في 2.4 ، حيث المنهجية المقترحة تستخدم ذاكرة قصيرة المدى متعددة الطبقات (LSTM) لربط أو تشفير سلسلة الدخل إلى شعاع ذي بعد ثابت ، ثم LSTM أخرى لفك تشفير سلسلة الهدف أو الخرج من الشعاع السابق الشكل 4.4 (fig 4.4) يوضح كيفية ذلك ، حيث كل مربع يعبر عن وحدة LSTM و <EOS> هو ما يقابل الرمز <END> كما رأينا في 3.3 .

أوائل استخدامات هذه البنية كانت في الترجمة من الإنكليزية للفرنسية ، ولكن تم تكييف هذه الشبكة لتصبح قابلة للاستخدام مع البيانات المستخدمة في المشروع.

إن سبب اختيار هذه البنية كان : (1) القدرة على التعامل مع دخل تسلسلي ذو طول غير معلوم مسبقا (2) نتيجة ما تم استنتاجه في [8,9] حيث وجد أن الشبكة تعلمت تمثيلات (Representation) معقولة للجمل التي تكون حساسة لترتيب الكلمات وثابتة للجمل المبنية للمعلوم و المبنية للمجهول (3) القدرة على النقاط كلاً من الهياكل الدلالية والنحوية للجمل.

لجعل الشبكة قادرة على إعطاء نتائج مرضية ، تم عكس تسلسلات دخل التدريب ، حيث تم استنتاج في [9] أن هذه الخدعة تنتج عن إدخال العديد من الاعتماديات قصيرة المدى (Short Term Dependencies) إلى مجموعة البيانات، حيث عندما نربط جملة الدخل بجملة الهدف ، فإن كل كلمة في الجملة الدخل تكون بعيدة عن الكلمة المقابلة لها في الجملة الهدف، من خلال عكس الكلمات في جملة الدخل ، لا يتغير متوسط المسافة بين الكلمات المقابلة في جملة الدخل و جملة الهدف. تم تطبيق هذه الخدعة في المشروع.

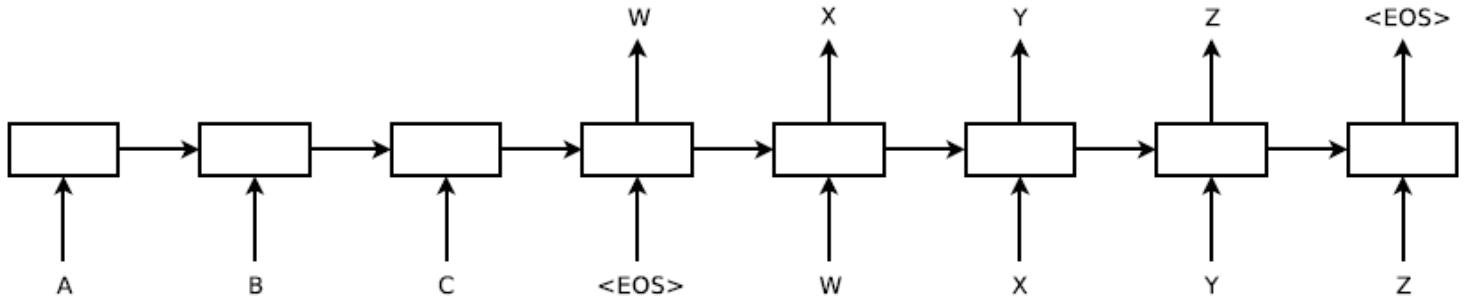


Fig 4.4

#### 4.4 التدريب بمنهجية إجبار المعلم (Training Using Teacher Forcing Method):

هذه التقنية تم اقتراحها في [10] حيث تعتبر نوع من أنواع التعلم المتكرر في الوقت الحقيقي (Real-Time Recurrent Learning - RTRL), حيث يتم استبدال خرج الشبكة التكرارية (Recurrent Network) بخرج الهدف أو المعلم (Teacher Signal) وذلك لأننا نقوم بالتدريب بإشراف أو بمعلم , ويتم ادخال هذه الإشارة كدخل للشبكة في المرحلة اللاحقة .

هذه الحركة تهدف في [10] لجعل بيانات التدريب لها دور أكثر أهمية مما سبق و تقوم بجعل الشبكة ديناميكية أكثر بحيث لا يوجد نمط ثابت يمثلها. يؤدي هذا إلى ظهور تمثيلات داخلية ديناميكية يتم توزيعها ، بمعنى ما ، في كل من المكان والزمان.

هذه المنهجية تم تطبيقها أيضا في [9,8] حيث تم ملاحظة أن الشبكة تتقارب (Converges) بشكل أسرع نحو خطأ أقل.

#### 5.4 مرحلة التدريب و الفحص (Training and Testing):

تم أخذ 80% من البيانات المستخدمة (حيث الحجم الكلي هو 34253 عينة , ما يقابل 27402 عينة تدريب) كنماذج تدريب , وتم بناء الشبكة تشفير – فك تشفير حيث كان طول المشفر أي عدد الوحدات LSTM بطول أطول سلسلة دخل و جميع سلاسل الدخل تم تكييفها على نفس الطول الأعظمي وهو 75 و دخل كل وحدة هو شعاع التضمين ذو طول 300 , ذلك تم عن طريق حشو (Padding) للرمز <PAD> المذكور في 3.3 و بين كل سلسلة وسلسلة الرمز الفاصل <SEP>, و في نهاية السلسلة كاملة تم وضع الرمز <END>.

أما بالنسبة لطرف الخرج أي طرف فك تشفير في الشبكة فتمت المعالجة بطريقة مماثلة , أي بجعل شبكة فك تشفير بطول أطول سلسلة خرج وهو 61 و دخل كل وحدة هو شعاع التضمين ذو طول 300, وتم الحشو أيضا ووضع الرمز <START> في بداية السلسلة ولكن هذا تم من أجل إشارة المعلم (Teacher Signal) كما تم شرحه في 4.4.

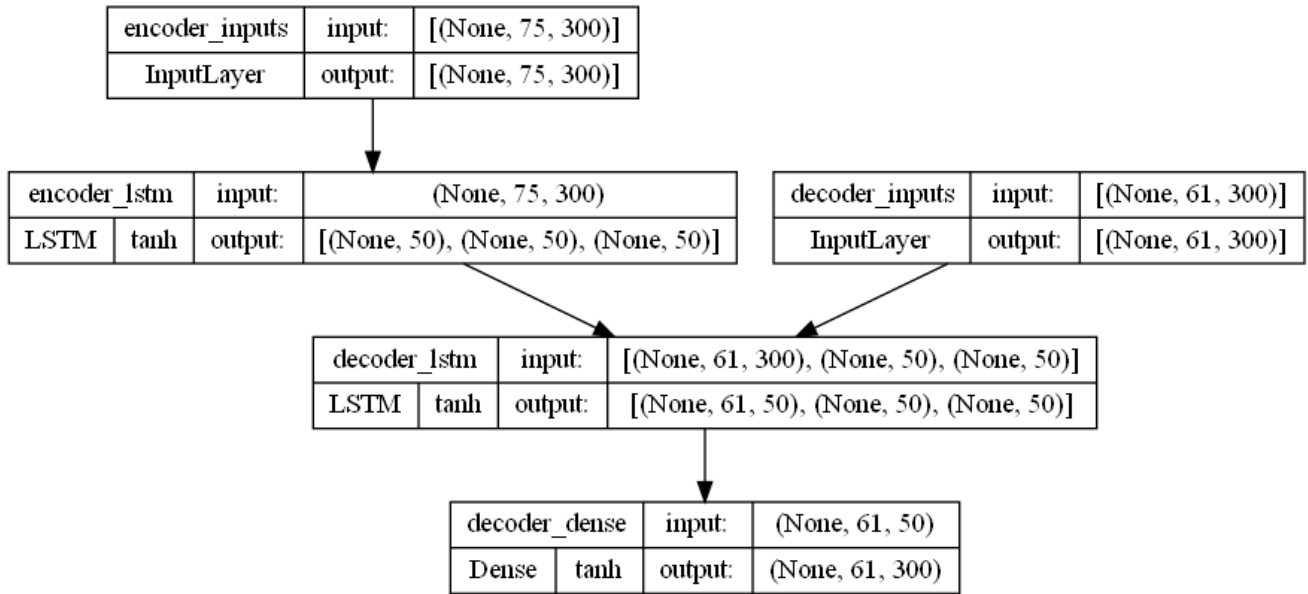
كمثال على ما قيل :

Encoder Input: ["field", "in", "cow", "<SEP>", "field", "in", "cow", "<SEP>", "field", "in", "cow", "<PAD>", "<PAD>", "<PAD>", ".....", "<PAD>", "<PAD>", "<PAD>", "<END>"]

Decoder Input (Teacher Signal): ["<START>", "th", "re", "e", "</w>", "cow", "s", "</w>", "are", "</w>", "in", "</w>", "the", "</w>", "field", "<PAD>", "<PAD>", "<PAD>", ".....", "<PAD>", "<PAD>"]

Decoder Output (Desired Output): ["th", "re", "e", "</w>", "cow", "s", "</w>", "are", "</w>", "in", "</w>", "the", "</w>", "field", "<PAD>", "<PAD>", "<PAD>", ".....", "<PAD>", "<PAD>"]

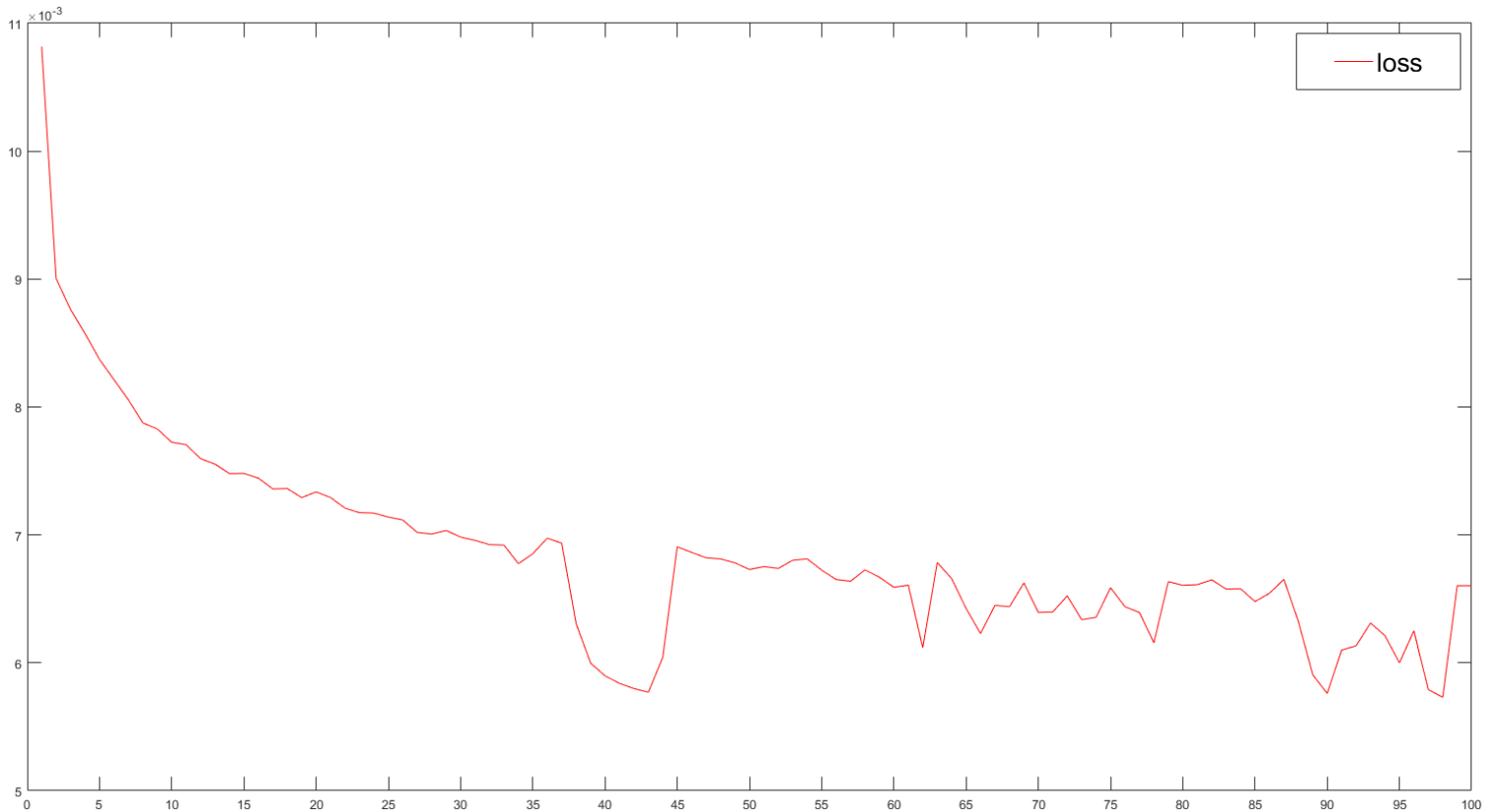
بنية الشبكة التي قد تم بنائها للتدريب موضحة بالشكل 5.4 (fig 5.4) .



**Fig 5.4**

إن مخرجات المشفر هي حالته الأخيرة ألا وهي حالة بوابة in z و out z و CEC مع الشعاع الممثل للسلسلة وهو خرج المشفر الحقيقي , كل هذه القيم يتم تمريرها كمدخل لفاك التشفير ليتم بعدها توليد أشعة تضمين بطول 300 , حيث كل وحدة تخرج شعاع تضمين يدل على رمز الخرج (Output Token Embedding) , بهذه الطريقة تم جعل المشكلة أو تابع الهدف هو من نوع انحداري (Regression) , أي الهدف هو جعل الاوزان قادرة على إعطاء أشعة تضمين أقرب ما يكون لتلك التي في الهدف .

تم الوصول إلى دقة تدريبية قدرها Accuracy = %89.8 و %84.5 خسارة قدرها Loss = 0.00249 و ذلك بعد التدريب 200 مرة (Epochs) , الشكلان 6.4 و 7.4 يوضحان تغير كلا من الخسارة و الدقة مع الزمن.



**Fig 6.4**

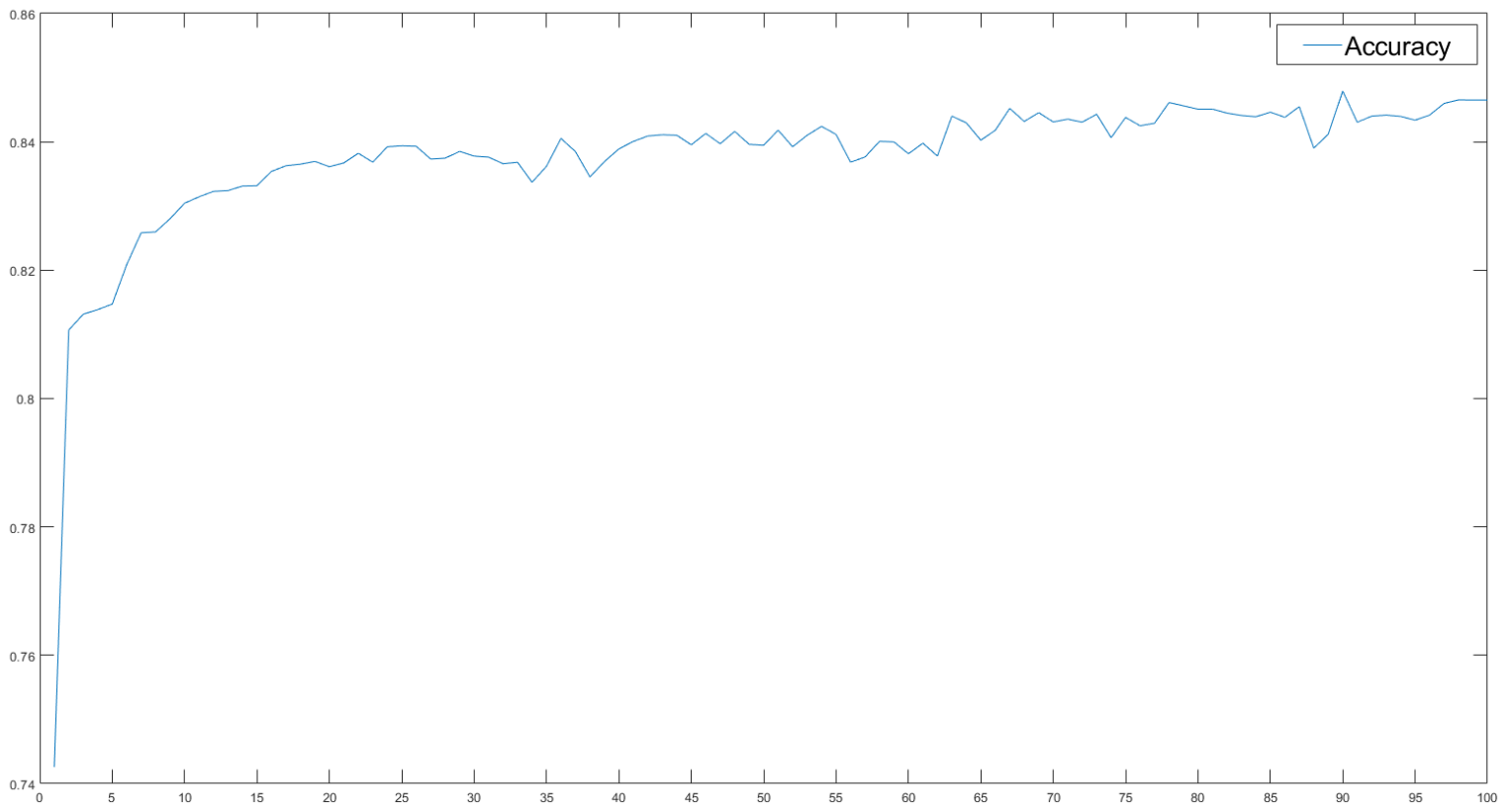


Fig 7.4

بسبب استخدام تقنية إجبار المعلم , فإن الشبكة المبنية للتدريب لا تصلح للفحص (Testing) , لأنه لن يكون هناك أي إشارة أصلا , لذلك تم بناء نموذج آخر للفحص ولكن باستخدام أوزان النموذج السابق طبعا , إن بنية نموذج الفحص مبينة بالشكل 7.4

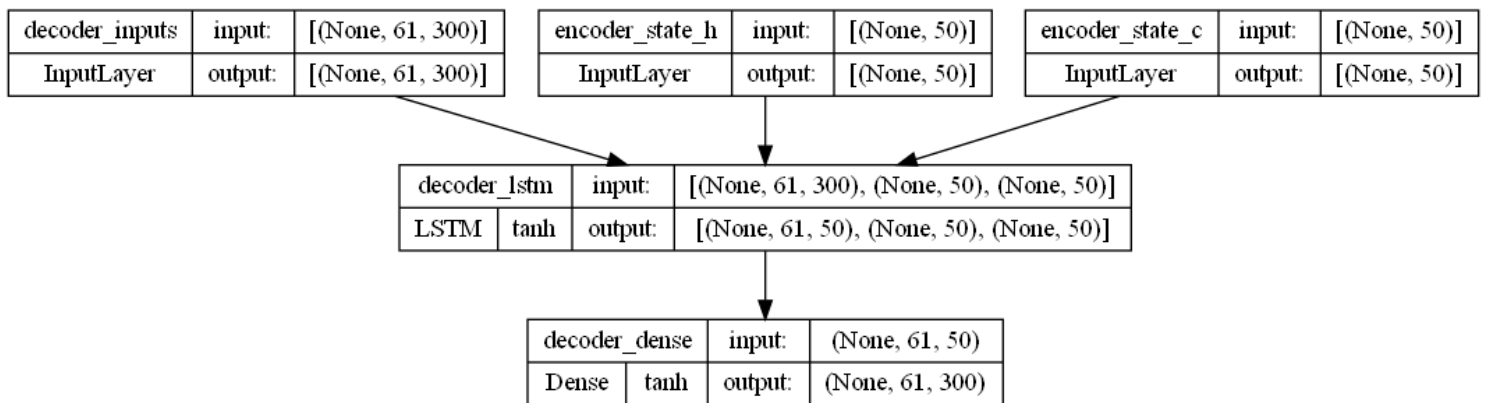


Fig 7.4

حيث أن بنية الشبكة مشابهة تماما للتدريب .

بلغت نسبة الدقة في مرحلة الفحص 74.4% وذلك تم على 20% من البيانات المتبقية والتي تعادل 6851 عينة فحص.

نستعرض بعض من هذه العينات :



### Example 1 :

Input : ['in', '</w>', 'field', '</w>', '<SEP>', '</w>', 'horse', '</w>', 'with', '</w>', 'horse', '</w>', '<SEP>', '</w>', 'horse', '</w>', 'in', '</w>', 'field', '</w>', '', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '.....', '<PAD>', '<PAD>', '<PAD>', '<END>']

Desired Output : ['two', '</w>', 'hor', 's', 'es', '</w>', 'in', '</w>', 'the', '</w>', 'field', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '.....', '<PAD>', '<PAD>', '<PAD>', '<END>']

Actual Output : ['a', 'airplane', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', 'ed', 'y', 'bag', 'er', '</w>', 'of', '<PAD>', '<PAD>', '</w>', 'of', '</w>', 'the', '</w>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>']

### Example 2 :

Input : ['on', '</w>', 'wall', '</w>', '', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '....', '<PAD>', '<PAD>', '<PAD>', '<END>']

Desired Output : ['gra', 'y', '</w>', 'hand', '</w>', 'ho', 'ld', '</w>', 'in', '</w>', 'a', '</w>', 'wall', '<PAD>', '<PAD>', '<PAD>', '.....', '<PAD>', '<PAD>', '<PAD>', '<END>']

Actual Output : ['a', 'ach', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', 'wall', '<PAD>', '<PAD>', 'bag', '<PAD>', 'ed', 'y', 'en', 'of', 'of', '</w>', 'bag', '</w>', 'bag', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>']

### Example 3 :

Input : ['throwing', '</w>', 'a', '</w>', 'ball', '</w>', '', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '....', '<PAD>', '<PAD>', '<END>']

Desired Output : ['p', 'it', 'ch', 'er', '</w>', 'tr', 'ow', 'ing', '</w>', 'a', '</w>', 'b', 'all', '<PAD>', '<PAD>', '<PAD>', '.....', '<PAD>', '<END>']

Actual Output : ['a', '</w>', 'skateboard', 're', '<PAD>', 'k', 'airplane', 'wearing', '</w>', 'of', '<PAD>', 'on', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<END>', '<PAD>', '<PAD>', '<PAD>', 'ed', '<PAD>', '<PAD>', 'ed', 'in', '</w>', 'of', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '<PAD>', '</w>', '</w>', 's', '</w>', 'with', '</w>', 's', '</w>', '</w>', '</w>', '</w>', '</w>', '</w>', '</w>', '</w>', '</w>', '<UNK>', '</w>', '</w>', '</w>', 'e', '</w>', '</w>', '</w>', '<UNK>', '</w>', '</w>']

هذه النتائج ليست بالمرضية أبداً و يرجح السبب على أنه يرتبط بشح البيانات التي تم إدخالها كدخل للشبكة كمواصفات الكائنات و تعدديتها و صفات العلاقات و تعدديتها أيضاً .

## الفصل الخامس

### الاستنتاجات و المقترحات

## الاستنتاجات و المقترحات (Conclusions and Suggestions)

1. يمكن توسيع أفق المشروع ليتم أتمتة اكتشاف الكائنات و العلاقات يما بينها , عن طريق بناء نموذج بتقانات الرؤية الحاسوبية قادرة على اكتشاف ذلك.
2. من المشاكل المعقدة في عملية حساب أشعة التضمين هي تحديد الطول الأمثل القادر على النقاط , تم البحث في ذلك في [11] و يمكن اقتراح تطبيق المنهجية المقترحة للحصول على الحد الأدنى من الطول المطلوب حسب الأبجدية البيانات المستخدمة.
3. إن النموذج الذي تم بنائه لم يحقق النتائج المرجوة و ذلك يرجع إلى شح في البيانات المستخدمة كدخل للنموذج , لذا يمكن اقتراح تعديل الدخل ليتضمن المعلومات المحتاجة.

- [1] Visual Genome Connecting Language and Vision  
Ranjay Krishna et al  
<https://arxiv.org/abs/1602.07332>      Dataset Website : <https://visualgenome.org>
- [2] Rico Sennrich , Barry Haddow and Alexandra Birch  
Neural Machine Translation of Rare Words with Subword Units  
<https://arxiv.org/abs/1508.07909v5>
- [3] Tomas Mikolov , Kai Chen , Greg Corrado and Jeffrey Dean  
Efficient Estimation of Word Representations in Vector Space  
<https://arxiv.org/abs/1301.3781>
- [4] Tomas Mikolov , Ilya Sutskever , Kai Chen , Greg Corrado and Jeffrey Dean  
Distributed Representations of Words and Phrases and their Compositionality  
<https://arxiv.org/abs/1310.4546v1>
- [5] word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding  
Yoav Goldberg and Omer Levy  
<http://arxiv.org/abs/1402.3722v1>
- [6] P. J. Werbos  
Backpropagation through time: what it does and how to do it  
doi: 10.1109/5.58337.
- [7] Sepp Hochreiter, Jürgen Schmidhuber  
Long Short-Term Memory.  
<https://doi.org/10.1162/neco.1997.9.8.1735>
- [8] Yoshua Bengio ,Kyunghyun Cho , Bart van Merriënboer , Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares and Holger Schwenk  
Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation  
<https://arxiv.org/abs/1406.1078>
- [9] Ilya Sutskever , Oriol Vinyals and Quoc V. Le  
Sequence to Sequence Learning with Neural Networks  
<http://arxiv.org/abs/1409.3215v3>
- [10] Ronald J. Williams and David Zipser  
A Learning Algorithm for Continually Running Fully Recurrent Neural Networks  
doi: 10.1162/neco.1989.1.2.270.

[11] Kevin Patel, Pushpak Bhattacharyya  
Towards Lower Bounds on Number of Dimensions for Word Embeddings  
<https://aclanthology.org/I17-2006/>

المواقع:

- [1] Stemming and Lemmatization  
<https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>
- [2] Should You Stem and Lemmatize  
<https://stackoverflow.com/questions/71261467/should-you-stem-and-lemmatize>
- [3] Byte Pair Encoding  
<https://leimao.github.io/blog/Byte-Pair-Encoding/>
- [4] Why Is Removing Stop Words not Always a Good Idea  
<https://medium.com/@limavallantin/why-is-removing-stop-words-not-always-a-good-idea-c8d35bd77214>
- [2] The Ultimate Guide To Different Word Embedding Techniques In NLP  
<https://www.kdnuggets.com/2021/11/guide-word-embedding-techniques-nlp.html>
- [3] NLP: Word Embedding Techniques Demystified  
<https://towardsdatascience.com/nlp-embedding-techniques-51b7e6ec9f92>
- [4] Word Embedding Techniques Word2Vec and TF-IDF Explained  
<https://tinyurl.com/2kdnc85d>
- [5] Word Embeddings CBOW vs Skip-Gram  
<https://tinyurl.com/2c5wsvux>
- [6] Input and output shape in LSTM (Keras)  
<https://www.kaggle.com/code/shivajbd/input-and-output-shape-in-lstm-keras/notebook>
- [7] Tensorflow Documentation  
[https://www.tensorflow.org/api\\_docs/python](https://www.tensorflow.org/api_docs/python)
- [8] Keras Team on GitHub  
<https://github.com/keras-team/keras/issues>
- [9] MongoDB Documentation  
<https://www.mongodb.com/docs/manual/reference>
- [10] MongoDB Community  
<https://www.mongodb.com/community/forums>

## الملحق

### ملحق أ : ملحق الاستعلامات (Queries)

سنستعرض في هذا الملحق , الاستعلامات التي تم برمجتها لبناء قاعدة البيانات المذكورة في الفصل الثاني.

```
// testing
```

```
db.tess.aggregate([ { $lookup: { from: "object_synsets", localField: "synset", foreignField:
"_id", as: "values" } }, { $project: { "values.values": 1, _id: 0 } }, { $unwind: "$values" },
{ $replaceWith: "$values" }
])
```

```
// join for synsets in images
```

```
db.images.aggregate([ { $match: { _id: ObjectId("6278f04627d868b8d636ac52") } }, { $lookup: {
from: "object_synsets", localField: "objects.synsets", foreignField: "_id", as: "values" } }, {
$project: { "values.values": 1, _id: 0 } }, { $unwind: "$values" },
{ $replaceWith: "$values" }
])
```

```
// join between attributes and objects (images)
```

```
db.images.aggregate([ {
  $unwind: "$objects"
},
{
  $project: {
    "objects.merged_object_ids": 1,
    "objects.object_id": 1,
    "image_id": 1
  }
},
{
  $lookup: {
    from: "attributess",
    let: {
      localId: "$objects.object_id",
      localImage: "$image_id"
    },
    pipeline: [
      { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
      { $unwind: "$attributes" },
      {
        $match: {
          $expr: {
```

```

        $eq: [
            "$attributes.object_id",
            "$$localId"
        ]
    }
}
    }
},
    ],
    as: "x"
}
},
{
    $lookup: {
        from: "attributess",
        let: {
            localId: "$objects.merged_object_ids",
            localimage: "$image_id"
        },

        pipeline: [
            { $match: { $expr: { $eq: ["$image_id", "$$localimage"] } } },
            { $unwind: "$attributes" },

            {
                $match: {
                    $expr: {

                        $in: [
                            "$attributes.object_id",
                            "$$localId"
                        ]
                    }
                }
            }
        ],
        as: "y"
    }
},
{
    $project: {
        "x": 1,
        result: {
            $reduce: {
                input: "$y.attributes.attributes",
                initialValue: [],
                in: { $concatArrays: ["$$value", "$$this"] }
            }
        }
    }
}

```



```

    }
  },
  {
    $project: {
      attributes: {
        $reduce: {
          input: "$x.attributes.attributes",
          initialValue: "$result",
          in: { $concatArrays: ["$$value", "$$this"] }
        }
      }
    }
  }
}

])

db.relationshipss.aggregate([
  { $sample: { size: 1001 } },
  {
    $unwind: "$relationships"

    {

      $lookup: {
        from: "attributess",
        let: {
          localObject: "$relationships.object.object_id",
          localImage: "$image_id",
        },

        pipeline: [
          { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
          { $unwind: "$attributes" },

          {
            $match: {
              $expr: {
                $eq: [
                  "$attributes.object_id",
                  "$$localObject"
                ]
              }
            }
          }
        ],
      },
    }
  },

```

```

        { $lookup: { from: "attribute_synsets", localField: "attributes.synsets",
foreignField: "_id", as: "attributes.synsets" } }

    ],
    as: "relationships.object.object_id"

}

}, { $lookup: { from: "object_synsets", localField: "relationships.object.synsets",
foreignField: "_id", as: "relationships.object.synsets" } },
{
    $lookup: {
        from: "attributess",
        let: {
            localObject: "$relationships.object.object_id",
            localImage: "$image_id",
            mergedObject: { $ifNull: ["$relationships.object.merged_object_ids", []] },

        },

        pipeline: [
            { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
            { $unwind: "$attributes" },

            {
                $match: {
                    $expr: {

                        $in: [
                            "$attributes.object_id",
                            "$$mergedObject"
                        ]
                    }
                }
            },
            {
                { $lookup: { from: "attribute_synsets", localField: "attributes.synsets",
foreignField: "_id", as: "attributes.synsets" } }

            ],
            as: "relationships.object.merged_object_ids"

        }
    },
    {
        $lookup: {
            from: "attributess",

```

```

let: {
  localSubject: "$relationships.subject.object_id",
  localImage: "$image_id",
},

pipeline: [
  { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
  { $unwind: "$attributes" },

  {
    $match: {
      $expr: {
        $eq: [
          "$attributes.object_id",
          "$$localSubject"
        ]
      }
    }
  },
  { $lookup: { from: "attribute_synsets", localField: "attributes.synsets",
foreignField: "_id", as: "attributes.synsets" } }

],
as: "relationships.subject.object_id"

}

}, { $lookup: { from: "object_synsets", localField: "relationships.subject.synsets",
foreignField: "_id", as: "relationships.subject.synsets" } }

,
{
  $lookup: {
    from: "attributess",
    let: {
      localObject: "$relationships.object.object_id",
      localSubject: "$relationships.subject.object_id",
      localImage: "$image_id",
      mergedObject: { $ifNull: ["$relationships.object.merged_object_ids", []] },
      mergedSubject: { $ifNull: ["$relationships.subject.merged_object_ids", []] }
    },

    pipeline: [
      { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
      { $unwind: "$attributes" },

      {
        $match: {

```

```

        $expr: {
            $in: [
                "$attributes.object_id",
                "$$mergedSubject"
            ]
        }

    },
    { $lookup: { from: "attribute_synsets", localField: "attributes.synsets",
foreignField: "_id", as: "attributes.synsets" } }

    ],
    as: "relationships.subject.merged_object_ids"

}

},
{ $lookup: { from: "relationship_synsets", localField: "relationships.synsets",
foreignField: "_id", as: "relationships.synsets" } }

, {
    $group: {
        _id: "$_id",
        relationships: { $push: "$$ROOT" },
        image_id: { $first: "$image_id" }
    }
},
{ $out: { db: "visual_genome", coll: "Results" } }

], { "allowDiskUse": true })

```

//testing

```

db.region_graphss.aggregate([

    {
        $unwind: "$regions"
    },

    {
        $project: {
            relation: "$regions.relationships.relationship_id",
            image_id: 1
        }
    },
    {
        $project: {

```

```

        relation: 1,
        image_id: 1
    }
},
{
    $lookup: {
        from: "Results",
        let: {
            localrelationship: "$relation",
            localImage: "$image_id",
        },

        pipeline: [
            { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
            { $unwind: "$relationships" },

            {
                $match: {
                    $expr: {
                        $in: [
                            "$relationships.relationships.relationship_id",
                            "$$localrelationship"
                        ]
                    }
                }
            },
        ],
        as: "x"
    }
},
{
    $group: {
        _id: "$_id",
        relationships: { $push: "$$ROOT" },
        image_id: { $first: "$image_id" }
    }
}

], { "allowDiskUse": true })

db.region_graphss.aggregate([
    {
        $sample: { size: 10 }
    },
    {

```

```

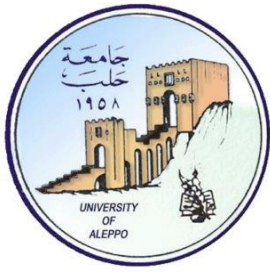
    $unwind: "$regions"
  },
  {
    $project: {
      regions: 1,
      _id: 0,
      image_id: 1
    }
  },
  {
    $lookup: {
      from: "Results",
      let: {
        localrelationship: { $ifNull: ["$regions.relationships.relationship_id", []] },
        localImage: "$image_id",
      },
      pipeline: [
        { $match: { $expr: { $eq: ["$image_id", "$$localImage"] } } },
        { $unwind: "$relationships" },
        {
          $match: {
            $expr: {
              $in: [
                "$relationships.relationships.relationship_id",
                "$$localrelationship"
              ]
            }
          }
        },
      ],
      as: "regions.relationships"
    }
  }, { $out: { db: "visual_genome", coll: "tesst" } }
], { "allowDiskUse": true })

```

## Summary

In this work, a system was created that generates texts descriptions of the content in a set of texts that contain relationships between objects using techniques from artificial intelligence and natural language processing, this was done by creating a database containing all the necessary information, and then adapting this data to remove any confusion that may harm the progress of the process, express the texts using neural technology, and then go to create a neural network capable of doing what is required.

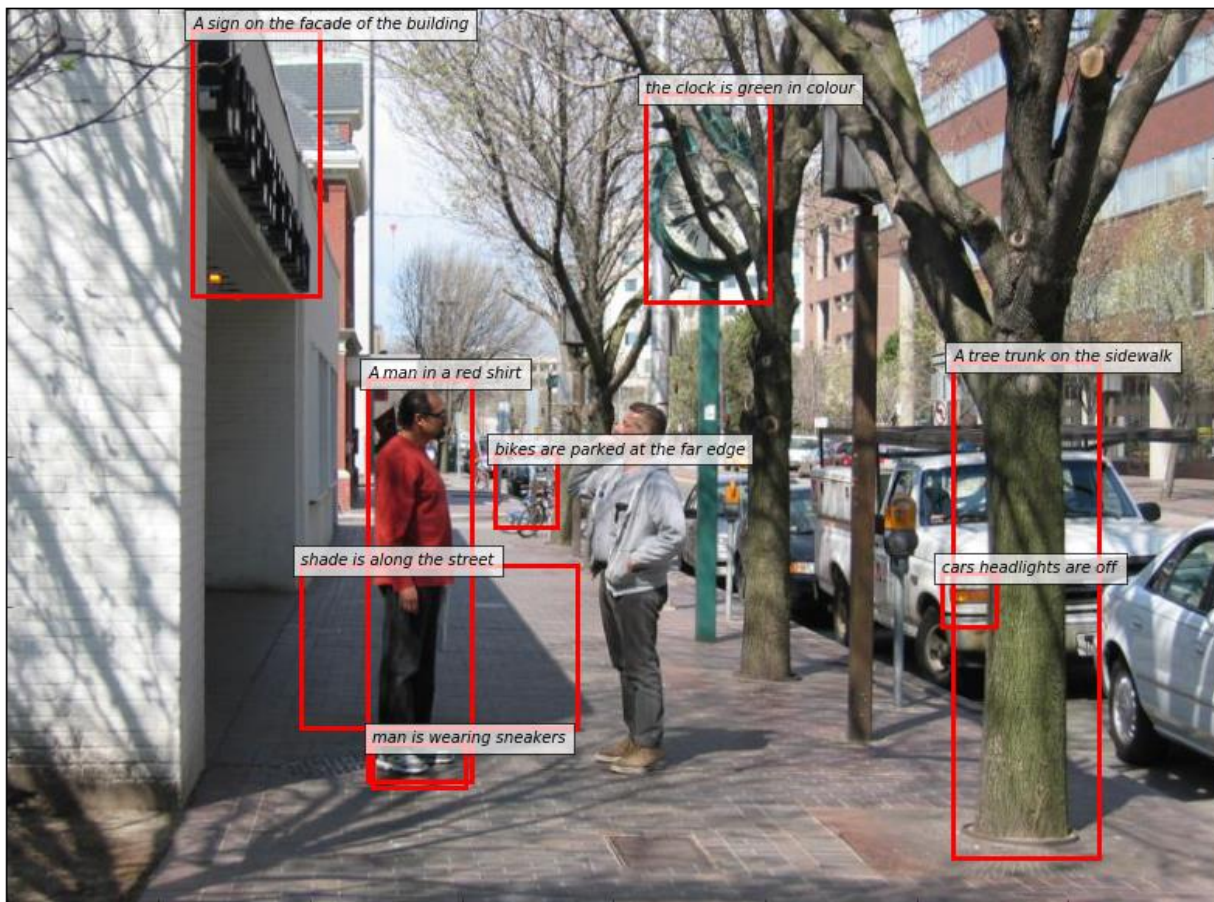
The aim of this work is to extract the hidden content through short sentences that clarify the relationships between the objects represented in the image corresponding to the short sentences. The project provides the ability to describe in general what is happening in an image. The project can be applied in the field of web and web pages to be specific to generate descriptions of images that explain them and put the explanation under the image in case the download speeds are slow or the hard disk space is limited.



University of Aleppo  
Faculty of Informatics  
Artificial Intelligence Dept.  
Fourth Year



## ***Detecting Context of Text With Its Respective Images Using The Semantic Web***



### Supervisor

Dr Fadel Sukkar

### Students

Ghaith Chamaa

Nihad Nabulse

Ali Deeb

Jodie Ali

*April 2022*  
*Fourth Year Project*