Difference between match and match_phrase on single word

Trying to get a deeper understanding of what happens here : https://github.com/jurismarches/luqum/issues/23 (https://github.com/jurismarches/luqum/issues/23)

We modified luqum to use "match" for one word queries but this seems to give inconsistent results with previous versions.

Note: This notebook assume you have a running instance of elasticsearch on localhost, default port. It will create a small test index on it.

Having a test index

In [1]:

```
INDEX_NAME = "test-es-match-vs-match-phrase"
```

In [2]:

```
analysis_setting = {
    'analyzer': {
        'test_match_analyzer': {
            'type': 'custom',
            'char_filter': [],
            'filter': ['word_delimiter'],
            'tokenizer': 'lowercase',
        },
    },
}
```

In [3]:

```
# mapping
MAPPING = {
    "ana_obj":{
        'properties': {
            "ana_message" : {
                "type" : "string",
                "analyzer" : "test_match_analyzer",
                "position_increment_gap" : 100
            },
        },
    },
}

SETTINGS = {
    "index": {
        "number_of_shards":1,
        "analysis": analysis_setting,
    },
}
```

In [4]:

```
# configure indexes
from elasticsearch import Elasticsearch
from elasticsearch.exceptions import NotFoundError
es = Elasticsearch()
out = []
# clean
try:
    out.append(es.indices.delete(INDEX_NAME))
except NotFoundError:
    pass
out.append(
    es.indices.create(
        INDEX_NAME,
        body={
            "settings": SETTINGS,
            "mappings": MAPPING,
        },
    )
)
out
```

Out[4]:

```
[{'acknowledged': True}, {'acknowledged': True}]
```

In [5]:

```
# helper to query
def search(query):
    result = es.search(
        index=INDEX_NAME,
        body={"query" : query.to_dict()})
    return [r['_id'] for r in result['hits']['hits']]
```

Testing

In [6]:

```
# add to index
es.index(index=INDEX_NAME, doc_type='ana_obj', id=1, body={"ana_message":"leading bio-technologies"}, refresh=True)
es.index(index=INDEX_NAME, doc_type='ana_obj', id=2, body={"ana_message":"leading market"}, refresh=True)
es.index(index=INDEX_NAME, doc_type='ana_obj', id=3, body={"ana_message":"bio-tech market"}, refresh=True)
None
```

In [7]:

```
from elasticsearch_dsl import query as q
```

In [8]:

```
search(q.Match(ana_message="market"))
```

Out[8]:

```
['2', '3']
```

In [9]:

```
search(q.MatchPhrase(ana_message="market"))
```

Out[9]:

```
['2', '3']
```

> so far, so good, for single word, match and match_phrase are the same.

```
In [10]: search(q.Match(ana_message="bio-technologies"))
Out[10]: ['1', '3']
```

```
In [11]: search(q.MatchPhrase(ana_message="bio-technologies"))
Out[11]: ['1']
```

> compounds words leaves to different results

```
In [12]: search(q.Match(ana_message="bio technologies"))
Out[12]: ['1', '3']
```

```
In [13]: search(q.MatchPhrase(ana_message="bio technologies"))
Out[13]: ['1']
```

> this is the analyzer, that make compounds words behaviour similar to having two separate words

Partial Conclusion

The difference in the match comes from the fact that compound words are separated. Then the meaning of match is not the same.

What is the behaviour of querystring, which is the one of reference for us.

```
In [14]: search(q.QueryString(query="ana_message:bio-technologies"))
Out[14]: ['1', '3']
```

```
In [15]: search(q.QueryString(query='ana_message:"bio-technologies"'))
Out[15]: ['1']
```

Conclusion

If we want to mimic `query_string` in luqum, we have to use `match` for words and `match_phrase` for phrases.